

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**CADESAÚDE: UM APLICATIVO PARA A PLATAFORMA ANDROID PARA
LOCALIZAR ESTABELECIMENTOS DE SAÚDE PÚBLICOS E PRIVADOS NO
ESTADO DE SANTA CATARINA**

JUCEMAR DIMON

Florianópolis/SC
2016/2

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO**

**CADESAÚDE: UM APLICATIVO PARA A PLATAFORMA ANDROID PARA
LOCALIZAR ESTABELECIMENTOS DE SAÚDE PÚBLICOS E PRIVADOS NO
ESTADO DE SANTA CATARINA**

JUCEMAR DIMON

Trabalho de conclusão de curso apresentado
como parte dos requisitos para obtenção do
grau de Bacharel em Sistemas de Informação
da Universidade Federal de Santa Catarina.

Florianópolis/SC
2016

JUCEMAR DIMON

CADESAUDE: UM APLICATIVO PARA A PLATAFORMA ANDROID PARA
LOCALIZAR ESTABELECIMENTOS DE SAÚDE PÚBLICOS E PRIVADOS NO
ESTADO DE SANTA CATARINA

Trabalho de conclusão apresentado como
requisito para obtenção do grau de Bacharel,
no curso de Sistemas de Informação da
Universidade Federal de Santa Catarina, sob
apreciação da seguinte Banca Examinadora:

Prof^a. Dr^a. Christiane Anneliese Gresse Von Wangenheim (UFSC)

Prof. Dr. Maurício Floriano Galimberti (UFSC)

Prof^a. Dr^a. Patricia Vilain (UFSC)

Aprovado em 09 de dezembro de 2016

Prof. Dr. Raul Sidnei Wazlawick (UFSC)
Orientador

Dedico este trabalho a toda a população de Santa Catarina que precisa ou algum dia já necessitou de um atendimento médico ou hospitalar e, por muitas vezes, gastou muito tempo para localizar um simples telefone de contato ou endereço de uma clínica ou hospital que atendesse suas necessidades em um momento de enfermo.

AGRADECIMENTOS

Agradeço a todos de minha família pela força e apoio na empreitada que é a admissão e conclusão de um curso de nível superior em uma universidade pública federal. Quero agradecer em especial a todos os amigos, colegas e conhecidos que se propuseram a instalar o aplicativo em seus dispositivos e ao respaldo dado, pois muitas questões foram repensadas e melhoradas a partir destes.

RESUMO

Em Santa Catarina existe uma quantidade muito grande de estabelecimentos de saúde de diversos tipos, são clínicas especializadas, consultórios médicos, hospitais, postos de saúde, policlínica entre outras. Para conseguir informações sobre estes estabelecimentos, como um telefone por exemplo, é necessário recorrer a uma lista telefônica ou um catálogo de convênio médico que traz os estabelecimentos conveniados ou, então, a algum provedor de busca na internet que traz uma grande variedade de informações sobre estes estabelecimentos. Algumas delas, no pior dos casos, podem em algum momento tornarem-se desatualizadas e, devido a isto, podem não ser de utilidade para alguns usuários, principalmente em uma situação de urgência em saúde. Foi baseado neste contexto que se optou pela escolha do tema para este projeto de conclusão de curso. Ele envolveu a criação de uma solução para facilitar aos cidadãos e visitantes de Santa Catarina na busca por estabelecimentos de saúde dentro do nosso Estado. Utilizando-se dos dados dos estabelecimentos de saúde que estão na base federal do Cadastro Nacional dos Estabelecimentos de Saúde, o CNES, foi desenvolvido um aplicativo para *smartphones* que possuam a plataforma Android. A escolha pela base do CNES como fonte de dados foi baseada nas várias questões que garantem a veracidade dessas informações divulgadas pelo Ministério da Saúde, dentre elas, as diversas legislações que regem os procedimentos de cadastramento, podem garantir informações fidedignas sobre os estabelecimentos. O aplicativo desenvolvido possibilita a busca por estabelecimentos de saúde mais próximos de sua cidade ou nos arredores de sua posição geográfica atual. Este aplicativo também permite que o usuário faça a busca por um estabelecimento que possua um serviço, uma especialidade médica ou um profissional de seu interesse ou necessidade. Ele foi projetado para funcionar sem a necessidade de conexão com a internet e já se encontra disponível de forma gratuita na loja de aplicativos da Google.

Palavras-chave: CNES. Android. Saúde. Estabelecimentos de Saúde. Mobile.

ABSTRACT

In Santa Catarina state there is a very large amount of health facilities of various kinds. Like Specialized Clinics, Doctors, Hospitals, Health Centers, among others. To get more information about these establishments, such as a phone for example, it is necessary to use a phone book or a catalog of medical “Health plans “or else a search provider on the Internet that provides information on these establishments. Some of them outdated may take additional time in search and harm the user especially in a medical urgency. The Brazilian Ministry of Health has a National registration system of health facilities (CNES) and disseminates free its database publicly on their specific portal. A number of legislative emplacements require that states and their municipalities carry out the registration of such establishments and keep this information up to date and reliable. Thus, we have a single database at the Federal level of all independent establishments if they are public or private. This course completion project involved the creation of solution to make it easier for citizens of Santa Catarina to search for health facilities within the state. Using “data “from health facilities that are in Federal CNES base, because they are public free information and there is legislation that standardizes the registration and upgrade processes, we developed an application for smartphones that have the Android platform that enables the search for the nearest health facilities in your town or geographic area. This application also allows the user to search for a property that has a service, a particular medical specialty or a professional interest or need. It is designed to work without need for connection to the internet and is now available free in the Google app store.

Key words: CNES. Android. Health. Health Facilities. Mobile.

LISTA DE FIGURAS

Figura 1 - Portal de divulgação dos dados dos estabelecimentos.....	22
Figura 2 - Fluxo de dados dos estabelecimentos de saúde	23
Figura 3 - Fluxograma da contribuição no projeto	28
Figura 4 - Arquitetura da plataforma Android	32
Figura 5 - Página de detalhamento do estabelecimento	40
Figura 6 - Modelagem do banco de dados.....	41
Figura 7 - Diagrama de classes.....	42
Figura 8 - Classes que fazem a leitura de arquivos de texto.....	43
Figura 9 - Principais métodos da classe ConversorHtmlParaJava.....	44
Figura 10 - Árvore de links para navegação.....	45
Figura 11 - Site com problemas	46
Figura 12 - Diagrama de atividades	47
Figura 13 - Configuração do Estado a ser capturado.....	49
Figura 14 - Android Dashboards - Platform Versions	50
Figura 15 - Estrutura de diretórios: projeto Android X Emulador.....	53
Figura 16 - Classe gestora de conexões com o banco de dados.....	54
Figura 17 - A tela de apresentação do aplicativo	55
Figura 18 - Trecho de código fonte que inicia a tela de apresentação	56
Figura 19 - Tela de estabelecimentos favoritos.....	57
Figura 20 - Menu de configurações.....	58
Figura 21 - Código fonte da definição do estabelecimento como favorito	60
Figura 22 - Diretório no dispositivo onde é salvo o estabelecimento favorito.....	61
Figura 23 - Estrutura interna do arquivo de estabelecimentos favoritos.....	61
Figura 24 - método de execução em segundo plano	63
Figura 25 - Trecho de código que define a SearchView.....	64
Figura 26 - Layout do modelo de apresentação dos dados dos estabelecimentos ...	65
Figura 27 - Resultado do layout na interface gráfica	66
Figura 28 - Tela de configurações.....	68
Figura 29 - Declarações para geração do Layout da tela de configurações	70
Figura 30 - Tela de busca por GPS.....	71
Figura 31 - Filtros da tela de busca por GPS	73
Figura 32 - Aplicação de filtro sobre filtro	75

Figura 33 - Tela de busca por parâmetros	77
Figura 34 - Configuração da busca parametrizada	79
Figura 35 - Trecho de código que representa as consultas ao banco.....	81
Figura 36 - Tela de detalhes do estabelecimento.....	83
Figura 37 - Layout da visualização em abas da interface gráfica.....	84
Figura 38 - definição das Uses Feature no arquivo de manifesto	87
Figura 39 - Página do aplicativo na Google Play Store	88
Figura 40 - Feedback com relatório de erro	90
Figura 41 - Comparativo do consumo de memória RAM	93

LISTA DE QUADROS

Quadro 1 - Plataforma de nomes de versões e níveis de API.....	30
--	----

SUMÁRIO

1 INTRODUÇÃO	12
2 OBJETIVOS	15
2.1 OBJETIVO PRINCIPAL.....	15
2.2 OBJETIVOS ESPECÍFICOS	15
2.3 JUSTIFICATIVAS.....	15
3 FUNDAMENTAÇÃO TEÓRICA.....	17
3.1 JSOUP	17
3.2 WEB CRAWLER	17
3.3 PADRÕES DE PROJETO	18
3.3.1 Data access object	18
3.3.2 Singleton	19
3.4 CADASTRO NACIONAL DOS ESTABELECIMENTOS DE SAÚDE.....	20
3.5 ANDROID.....	26
3.5.1 Histórico do Android	26
3.5.2 Arquitetura do Android.....	31
4 ESTADO DA ARTE	35
4.1 SISTEMA MÓVEL DE GEOLOCALIZAÇÃO E GEOPROCESSAMENTO PARA LOCAIS DE ATENDIMENTO EM SAÚDE.....	35
4.2 MEDSQUARE	37
5 DESENVOLVIMENTO DO PROJETO	39
5.1 A APLICAÇÃO DE CAPTURA DE DADOS	39
5.2 O APLICATIVO PARA DISPOSITIVOS ANDROID	49
5.2.1 O banco de dados	51
5.2.2 A tela inicial	54
5.2.3 A tela de favoritos.....	56
5.2.4 A tela de configurações	67

5.2.5 A tela de busca por GPS	71
5.2.6 A tela de busca parametrizada	76
5.2.7 A tela de visualização do estabelecimento	83
5.2.8 A publicação na loja de aplicativos	86
5.2.9 Avaliação do aplicativo	88
5.2.10 Análise geral do uso da memória	91
6 CONCLUSÕES	95
6.1 TRABALHOS FUTUROS	96
6.1.1 Melhorias na ferramenta de captura de dados	96
6.1.2 Adaptação do aplicativo para o funcionar parcialmente online	96
6.1.3 Adaptação do aplicativo para o funcionar totalmente online	97
6.1.4 Reengenharia da interface gráfica	98
REFERÊNCIAS	99
APÊNDICES	104
APÊNDICE A - ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE	105
APÊNDICE B - CODIGO FONTE DO PROGRAMA DE CAPTURA DE DADOS	109
APÊNDICE C - CODIGO FONTE DO APLICATIVO ANDROID	204
APÊNDICE D - ARTIGO SOBRE O TCC	548
ANEXOS	564
ANEXO A - TIPOS DE ESTABELECIMENTOS/UNIDADES	565
ANEXO B - TIPOS DE SERVIÇOS ESPECIALIZADOS	569

1 INTRODUÇÃO

Com a difusão e popularização dos sites de anúncios para empresas na internet, muitos usuários quando necessitam encontrar alguma informação sobre algum estabelecimento comercial em algum ponto da cidade podem preferir buscar estas informações nestes repositórios ao invés de utilizar as antigas listas telefônicas de papel. Atualmente, quando se faz uma busca por uma clínica médica ou hospital em algum motor de busca na internet, o que se pode receber como resposta em alguns casos é uma infinidade de resultados. Dentre estes resultados poderemos achar: páginas do *MAPLink*¹, páginas do *Apontador*², do *GuiaMais*³, da *Telelistas*⁴, do *Foursquare*⁵ e até mesmo do *Google Mapas*⁶ e do *Google Meu Negócio*⁷, entre outras diversas, contendo telefones, endereços da empresa procurada ou similares a ela. Com o passar do tempo a empresa que anunciou seu negócio ou o usuário que cadastrou ou marcou um local em uma rede social, como o *Foursquare* por exemplo, podem perder seu interesse nessas informações. Isto pode ocasionar no esquecimento da atualização desses dados criados e assim essas informações podem tornar-se inúteis a quem necessita por estarem desatualizadas. De forma geral, no pior dos casos, pode-se perder tempo tentando encontrar a informação correta que se procura, como o telefone de contato, por exemplo. Situações como esta, por muitas vezes, podem forçar o internauta a gastar mais tempo do que o esperado, uma vez que tal recurso deveria servir para facilitar sua vida. Em uma situação emergencial quando se precisa buscar por um estabelecimento de saúde usando-se da metodologia descrita acima, a falta de informações precisas pode tornar a situação mais agravante.

As plataformas mobile tem crescido e se popularizado nos últimos anos. No ano de 2014 as vendas de dispositivos no mundo cresceram em torno de 23% (EXAME.COM, 2015). Atualmente os smartphones estão presentes no cotidiano de qualquer pessoa, pesquisas apontam que o seu uso ao longo do dia também tem

¹ <http://maplink.com.br/>

² <http://www.apontador.com.br/>

³ <http://www.guiamais.com.br/florianopolis-sc>

⁴ <http://www.telelistas.net/>

⁵ <https://pt.foursquare.com/>

⁶ <https://www.google.com.br/maps>

⁷ <https://www.google.com.br/intl/pt-BR/business/>

aumentado cada vez mais (NAÍMA SALEH, 2015). Existe uma grande variedade de aplicações para auxiliar na resolução de problemas dos mais variados tipos e devido à mobilidade proporcionada por esses dispositivos a sua utilização não fica presa a um ambiente fixo, como acontece com os PCs. Baseando-se nesse contexto, desenvolvedores pesquisam sobre as dificuldades existentes na rotina da pessoa para projetar e promover soluções com o intuito de ganhar sua fatia no mercado de aplicativos para as plataformas IOS, Android e *Windows Phone*.

Muitas empresas de vários setores diferentes do mercado têm percebido a popularidade e a influência dos smartphones na vida dos seus clientes e vem apostando nessas tecnologias para ampliar seu canal de comunicação com seus consumidores (VOXAGE SERVIÇOS INTERATIVOS, 2016).

Empresas privadas prestadoras de serviços de saúde como clínicas especializadas, centros médicos integrados e hospitais tem apostado no mercado de aplicativos para as plataformas mobile como ferramenta útil para divulgação de seus serviços e de comunicação com seus clientes. Isto pode ser percebido em Clínica Privada de Especialidades Villa María (2015), em Social Advisors (2015) e em Xavier (2015).

Empresas de convênios médicos, odontológicos, e outros, investem na produção de aplicativos para smartphones para auxiliar seus clientes na localização de suas clínicas, filiais e profissionais conveniados (BUENO, 2016).

Empresas de software pesquisam as necessidades do mercado nas áreas da saúde e promovem soluções que são comercializadas, ou distribuídas livremente aos usuários. Um exemplo disto é o aplicativo *MedSquare* desenvolvido por WINTIMEIPC (2015) para a plataforma Android que é distribuído livremente na loja de aplicativos da Google.

Este projeto tem como foco o desenvolvimento de uma aplicação para dispositivos móveis compatível com a plataforma Android para auxiliar os cidadãos do estado de Santa Catarina na busca por uma clínica, hospital, laboratório ou consultório médico. Como base de informações para esse aplicativo serão utilizados dados do CNES (Cadastro Nacional dos Estabelecimentos de Saúde). Estes, por serem dados públicos disponibilizados oficialmente pelo Ministério da Saúde são informações fidedignas a respeito dos estabelecimentos. Existe uma grande quantidade de publicações de portarias e notas técnicas pelos órgãos controladores oficiais designando sobre a obrigatoriedade da manutenção destas informações, muitas delas

podem ser vistas em na seção de consulta à legislação em Cnes (2016). Sendo assim um dos grandes pontos positivos deste projeto é a confiabilidade dos dados disponibilizados no aplicativo.

2 OBJETIVOS

Neste capítulo serão apresentados todos os objetivos: principal e específicos do projeto desenvolvido e sua justificativa para comprovação de sua importância de seu desenvolvimento.

2.1 OBJETIVO PRINCIPAL

Desenvolver um aplicativo utilizando como base os dados públicos disponibilizados pelo Ministério da Saúde para fornecer uma forma prática e precisa de localizar um estabelecimento de saúde no Estado de Catarina.

2.2 OBJETIVOS ESPECÍFICOS

Outros objetivos foram eleitos para este projeto, dentre eles destacam-se:

1. Produzir um documento de especificação de requisitos do aplicativo de busca por estabelecimentos;
2. Produzir uma ferramenta de captura de dados dos estabelecimentos de saúde que tenha como produto final de sua execução um arquivo de banco de dados para o SGBD *SQLite*, o qual será utilizado com base de consulta para o aplicativo;
3. Produzir um aplicativo para smartphones para consulta de estabelecimentos de saúde;
4. Publicar o aplicativo para consulta de estabelecimentos de saúde na loja de aplicativos da Google.

2.3 JUSTIFICATIVAS

Ao realizar uma busca na loja de aplicativos do Google por opções que possibilitem aos usuários a localização de locais que prestam atendimento em saúde recebe-se muitas opções como resposta. Elas vão desde aplicações específicas para clientes de clínicas e convênios, como pode ser visto em Portal Unimed Ltda (2015), a simples aplicações para localizar locais que possuam atendimentos emergenciais,

como ocorre em DEVDECISION CONSULTORIA EM TECNOLOGIA (2015). Cada um desses aplicativos limita a quantidade de estabelecimentos mostrados. Podem limitar devido a alguns não serem conveniados com a empresa a qual pertence o aplicativo. Também limitam devido à necessidade de os próprios usuários serem os responsáveis pela alimentação da base de dados dos aplicativos. Pode ocorrer casos em que o aplicativo não é muito popular, sendo assim os usuários não contribuirão e a base de dados torna-se pequena.

O portal do DATASUS disponibiliza uma página onde é possível qualquer cidadão fazer uma busca por estabelecimentos cadastrados. Esses dados são públicos, porém pouco divulgados para a sociedade e suas páginas de busca possuem uma interface pouco muito amigável, uma vez que não seguem os padrões definidos pelo *World Wide Web Consortium*⁸(também conhecido como W3C). Pensando em uma forma de auxiliar esses usuários que precisam localizar um local que preste atendimento de saúde de forma rápida e precisa somado a praticidade e mobilidade fornecida pelos aplicativos mobile, definiu-se a criação de um aplicativo utilizando os dados do CNES como base de dados para consulta.

Espera-se que o projeto possibilite a produção de uma ferramenta que seja de utilidade pública para a população e que ela faça parte da rotina do cidadão do nosso estado.

⁸ <http://www.w3.org>

3 FUNDAMENTAÇÃO TEÓRICA

São apresentados neste capítulo alguns conceitos, técnicas e ferramentas que auxiliaram o autor na definição do tema e no desenvolvimento da solução projetada no objetivo deste trabalho.

3.1 JSOUP

Como afirma Jonathan Hedley (2015), *JSoup* é uma biblioteca Java que permite ao desenvolvedor manipular conteúdo HTML. Ela possui uma API (*Application Programming Interface*) que permite ao usuário extrair e manipular dados das páginas HTML utilizando DOM, CSS, e métodos do *jQuery-like*. Com ele é possível:

- Ler e converter HTML de uma URL, cadeia de URLs, arquivos ou *Strings*;
- Buscas dados das páginas HTML utilizando DOM ou seletores CSS;
- Manipular elementos HTML, atributos e textos;
- Organizar e formatar elementos HTML.

JSoup é um projeto *open source* distribuído livremente sob a MIT *license*⁹. O seu código fonte está disponível no *GitHub*¹⁰.

Para o projeto que foi desenvolvido, esta biblioteca Java foi utilizada na aplicação de captura de dados com o objetivo de conectar-se ao servidor do DATASUS para acessar as páginas do site do CNES para capturar suas *Tags* HTML que contém os dados dos estabelecimentos de saúde de interesse do projeto.

3.2 WEB CRAWLER

Um *web crawler* segundo Tavares e Lóscio (2013) é um agente de software que acessa a WEB de forma automatizada através de URLs para extrair dados. A esse processo dá-se o nome de *crawling*. O trabalho de Tavares, Lóscio e Oliveira (2012) complementa o conceito definindo que este componente automatizado visita as páginas HTML e navega pelo seu conteúdo capturando-o. Os links contidos nas

⁹ <http://jsoup.org/license>

¹⁰ <https://github.com/jhy/jsoup>

páginas são adicionados à sua fila de URLs que serão visitadas na sequência. Assim, um *web crawler* pode varrer toda a web de forma recursiva.

Foi baseado no conceito citado acima que se pensou em desenvolver uma aplicação específica que pudesse fazer a captura dos dados contidos nas páginas HTML do site do CNES. Neste projeto foi desenvolvido uma ferramenta de captura de dados nomeada de *JWebCrawlerCNES*. Esta ferramenta além de comportar-se como um *web crawler*, armazenando e visitando as URLs das páginas do site do CNES, realiza a seleção e captura dos dados das páginas cujas URLs visitadas utilizando a biblioteca JAVA *JSoup* e armazena esses dados em um banco de dados *SQLite* que também é criado pela aplicação utilizando a biblioteca Java *SQLite JDBC*.

3.3 PADRÕES DE PROJETO

Padrões de projeto podem ser vistos como uma solução que já foi testada para um problema. Desta forma, um padrão de projeto geralmente descreve uma solução ou uma instância da solução que foi utilizada para resolver um problema específico. Nas duas seções seguintes serão apresentados dois padrões que foram utilizados neste projeto.

3.3.1 Data access object

O padrão de desenvolvimento conhecido como DAO, do inglês *Data Access Object*, segundo Taborda (2015), é um padrão introduzido no ambiente JEE para simplificar e desacoplar a interação das aplicações Java com a API JDBC. Praticamente todas as aplicações corporativas possuem algum mecanismo de persistência e o mais amplamente utilizado são os bancos de dados e a linguagem SQL é a mais popular para promover a comunicação com os gerenciadores de bancos de dados, a API JDBC (*Java Database Connectivity*) foi desenvolvida justamente para resolver essa questão. Antes da utilização de JAVA os sistemas corporativos eram escritos em linguagens orientadas a processos que continha muitos códigos SQL misturados. Quando estes sistemas foram refeitos com JAVA, o padrão DAO tinha como objetivo a separação dessa grande quantidade de código SQL nos sistemas legados. No padrão DAO, todas as comunicações com o mecanismo de persistência são mediadas por um objeto escrito especificamente para mapear as informações dos

objetos específicos de transporte para a API de persistência e vice e vversa. Toda a lógica de mapeamento e execução das instruções é deixada dentro do objeto DAO, desta forma, isolando a aplicação da API de persistência por completo. O objeto DAO é criado para operar e interagir com o mecanismo de persistência em nome da aplicação. Ele, geralmente, executa os quatro tipos de operações básicas: Criar; Recuperar; Alterar e Apagar (CRUD - do inglês *Create, Retrive, Update , Delete*). As ações de edição são invocadas simplesmente passando um objeto contendo as informações que se deseja editar.

No projeto desenvolvido tal padrão foi utilizado tanto na aplicação de captura de dados quanto no aplicativo para Android. O padrão foi seguido durante a implementação das classes que manipulam os objetos das classes de modelos de dados como: Estabelecimentos; Profissionais; Serviços. Tudo isso para auxiliar na manipulação das operações dessas classes e suas respectivas representações como tabelas no banco de dados.

3.3.2 Singleton

Singleton é um tipo de padrão de projetos. É uma convenção que, se programada da forma correta, garante que um único objeto de uma classe de interesse será instanciado e que há apenas um ponto global de acesso a ela, ou seja, em qualquer ponto da aplicação de uma forma geral caso haja a intenção de uso do objeto cuja classe foi implementada usando este padrão, chama-se o construtor do objeto e caso ele já tenha sido criado, tem-se uma referência dele. Este tipo de padrão de projeto é muito útil quando os objetos possuem atributos de configuração que não podem se repetir. Um outro cenário onde ele é útil seria quando os objetos consomem muitos recursos do sistema, nesses casos evita-se ter mais de uma instância do mesmo na memória (FREEMAN et al., 2009, p. 132).

No projeto desenvolvido esse padrão foi utilizado na aplicação de captura de dados. O padrão foi seguido durante a implementação das classes que manipulam a comunicação com o banco de dados *SQLite* como um mecanismo para garantir uma correta manipulação dos processos de abertura e fechamento de conexões com o banco.

3.4 CADASTRO NACIONAL DOS ESTABELECIMENTOS DE SAÚDE

O Ministério da Saúde define que todos os estabelecimentos, públicos e privados, que prestam algum tipo de serviço em saúde ao ser humano devem ser cadastrados em sua base de dados federal (BRASIL. Portaria nº 511, Diário Oficial da União, 2000). Esse cadastro é efetuado mediante ao preenchimento pela empresa de um formulário através do qual são coletados todos os dados de sua estrutura física, organizacional e humana, e a sua análise e aprovação pelo órgão responsável pelo cadastro que, geralmente, é uma secretaria municipal de saúde. Após o cadastro aprovado, o estabelecimento é registrado e passa a ser identificado na base de dados através do código conhecido como CNES (Cadastro Nacional de Estabelecimentos de Saúde). Todas as empresas cuja a atividade principal ou secundária é o fornecimento de serviços relacionados com a saúde humana precisam ter cadastro no Ministério da Saúde e os dados que são usados para o cadastro precisam ser verídicos e atualizados.

O cadastro nacional de estabelecimentos de saúde foi instituído pela Portaria nº 376, Diário Oficial da União (2000). Foi publicada no Diário oficial da União e permaneceu em consulta pública durante alguns meses onde os gestores municipais e estaduais e a população interessada contribuíram com suas experiências e sugestões. Após, esta consulta foi encerrada e definiu-se a Portaria nº 511, Diário Oficial da União (2000) que normatizaria o processo de cadastros dentro do território brasileiro. Todo esse sistema representa um desejo muito antigo de todos que utilizam as informações sobre saúde no país para seu trabalho diário. Os dados que são coletados sobre os estabelecimentos auxiliam os gestores públicos a visualizar o que há disponível em variedades de estabelecimentos no país para planejar suas ações operacionais e gerenciais na área da saúde (MINISTÉRIO DA SAÚDE, 2006, p. 5). Estes dados coletados dos estabelecimentos, segundo explicado na seção sobre CNES em DATASUS (2016) são:

- Informações básicas gerais como nome fantasia, razão social;
- Endereçamento/Localização como: logradouro, bairro, CEP;
- Gestor responsável pela atualização do cadastro como: Secretaria Municipal de Saúde ou Secretaria Estadual de Saúde;
- Atendimentos prestados como: internação, atendimento ambulatorial, atendimento emergencial;

- Caracterização da empresa como: natureza jurídica, tipo de retenção de tributos;
- Equipamentos instalados como: RX, Tomógrafo, ultrassom;
- Serviços de Apoio como: SAME, S. Social, Lavanderia;
- Serviços Especializados como: Cardiologia, Nefrologia, Oftalmologia;
- Instalações Físicas como: leitos, salas, consultórios;
- Dados dos profissionais como: CBO, Carga horária, forma de vínculo;
- Equipes existentes como: ESF (Equipe de Estratégia de Saúde da Família);
- Cooperativas.

O empreendimento é gigantesco e os dados coletados são utilizados como base para operacionalizar todos os sistemas de informação em saúde do Ministério da Saúde que são a base para uma boa gestão pública, além, é claro, de adquirir conhecimento efetivo de como está formado o universo de estabelecimentos que cuidam da saúde da nossa população, desde nos grandes centros, até nas mais longínquas localidades, tornando visível esse cenário a toda sociedade, proporcionando ao gestor conhecer a rede assistencial existente e sua potencialidade, imprescindíveis nos processos de planejamento em saúde, regulação, avaliação, controle e auditoria, bem como dar maior visibilidade ao controle social para o melhor desempenho de suas funções DATASUS (2009b, p. 1).

A implementação efetiva do sistema ocorreu em 2003, desde então ações para sua manutenção e evolução vêm sendo tomadas para que o projeto mantenha uma roupagem mais moderna e amigável. Para tal, frequentemente, novas portarias vêm sendo emitidas pelo Ministério da Saúde para reforçar a importância das ações de cadastramento e atualização destas bases de dados, sempre lembrando das responsabilidades de cada parte e das devidas penalidades (BRASIL. Portaria nº 1.646, Diário Oficial da União, 2015). A criação de uma base de dados única e federal de todos os estabelecimentos de saúde do país é uma tarefa grande que deve envolver todas as esferas da administração pública Federal, Estadual e Municipal para que se possa manter uma base de dados com informações fidedignas para sistematicamente providenciar sua disseminação perante o território nacional. A Figura 1 exibe o portal principal utilizado para disseminar tais informações para a sociedade brasileira.

Figura 1 - Portal de divulgação dos dados dos estabelecimentos

Ministério da Saúde
CNESNet
 Secretaria de Atenção à Saúde
 DATASUS

Estabelecimento de Saúde

Identificação
 CADASTRO NO CNES EM: 26/7/2004 ÚLTIMA ATUALIZAÇÃO EM: 19/9/2013 DATA DE ATUALIZAÇÃO LOCAL: 19/9/2013

Nome: HOSPITAL UNIVERSITÁRIO **CNPJ:** 07.971.745 **CPF:** 07.971.745
Nome Empresarial: UNIVERSIDADE FEDERAL DE SANTA CATARINA **Personalidade:** JURÍDICA
Legislação: CAMPUS UNIVERSITÁRIO **Regime:** 0/0 **Telefone:** (41) 3722-2574

Complemento: Rua: 8804-000 **Cidade:** FLORIANÓPOLIS - SC **CEP:** 88040-000 **UF:** SC

Tipo Estabelecimento: HOSPITAL GERAL **Sub-Tipo Estabelecimento:** FEDERAL **Dependência:** GERAL

Natureza da Organização: ADMINISTRAÇÃO PÚBLICA - AUTARQUIAS **Dependência:** GERAL

Número Abast: 1774-023-181-03-POV **Órgão Expediente:** SUS **Data Expediente:** 24-05-2013

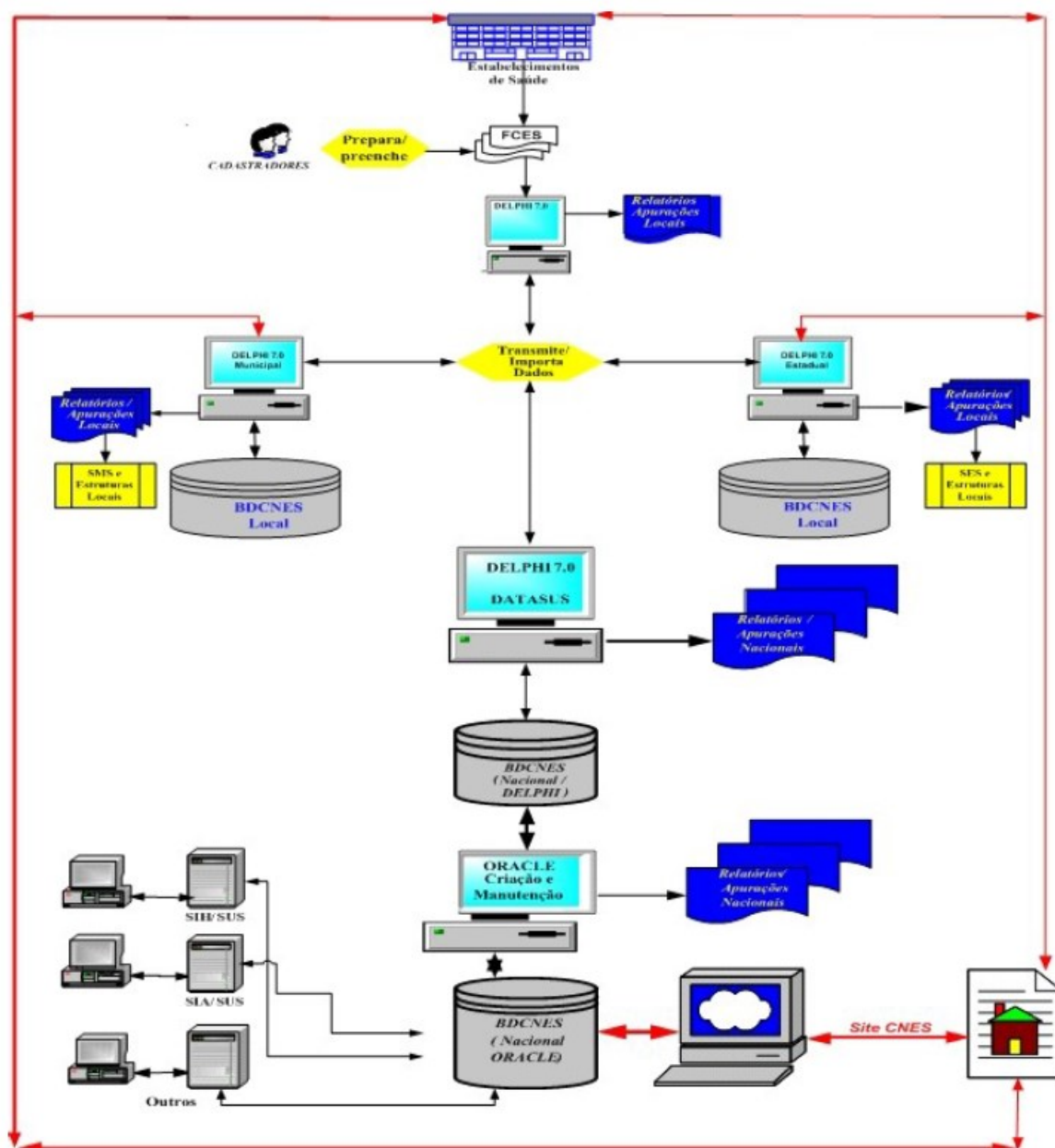
Módulos:

Basic	Código	Ambulatório	Hospitalar	Maternidade	Profissionais
Habitacões	Prótese Coração	Centro de Geriá	Neonatal	Ecografia	Respirat e Toraxoscopia
Teleatendimento	Org. Funcional	Sol. Amb. (Kerato)			

Fonte: DATASUS (2015b)

Os responsáveis pela localização, cadastro, inspeção e atualização dos cadastros dos estabelecimentos de saúde dentro desse sistema são as secretarias de saúde municipais ou estaduais. Elas podem criar e gerenciar seu próprio fluxo de trabalho, porém estas devem sempre preconizar as normas e prazos estipulados pelas portarias, notas técnicas e normativas e as leis emitidas pelas esferas superiores. A Figura 2 abaixo apresenta o fluxo oficial de todos os processos que envolvem o sistema (DATASUS, 2009a, p. 3):

Figura 2 - Fluxo de dados dos estabelecimentos de saúde



Fonte: DATASUS (2009a, p. 9)

As secretarias municipais de saúde são as responsáveis pelo cadastramento dos estabelecimentos de saúde lotados em seus municípios. Elas devem buscar por estes estabelecimentos, solicitar a documentação necessária e promover seus cadastramentos em seu sistema de informação SCNES. Este software foi desenvolvido pelo DATASUS para que os gestores municipais e estaduais desses estabelecimentos pudessem enviar os dados dessas empresas de forma digital e eletrônica para o ministério da saúde, evitando assim mais burocracia e geração

desnecessária de formulário de papel. O software SCNES¹¹ pode ser adquirido pelo site do DATASUS e é de uso da administração pública, porém como é de livre distribuição na internet é possível a qualquer cidadão fazer o download e instalar em seu computador pessoal em sua casa, porém durante o processo de instalação é exigido que se crie um usuário e se escolha um perfil para este usuário. Segundo DATASUS (2009a, p. 26), cada perfil habilita funcionalidades diferentes. Alguns dos perfis disponíveis para a escolha são:

- Estabelecimento: onde o usuário será o próprio estabelecimento de saúde que pode usar o software para enviar suas informações para cadastro ou alteração para o órgão que é responsável pela gestão do seu estabelecimento, que geralmente é uma secretaria municipal de saúde;
- Secretaria Municipal de Saúde: quem escolhe esse perfil geralmente é uma secretaria municipal de saúde que é gestora dos estabelecimentos lotados em seu município, sendo este de pequeno porte, porém não possui gestão plena sendo assim subordinado à Secretaria Estadual de Saúde, logo todas as informações sobre seus estabelecimentos são transmitidas de forma digital para a Secretaria Estadual e é esta quem faz o envio das informações ao Ministério da saúde;
- Secretaria Municipal de Saúde com Gestão Plena: Este perfil é indicado para municípios de grande porte. Neste caso todas as informações dos estabelecimentos são transmitidas de forma digital diretamente para o Ministério da Saúde;
- Secretaria Estadual de Saúde: Este perfil deve ser selecionado pela Secretaria Estadual de Saúde de cada Estado pois nele existe funcionalidades que permitem receber os dados que são transmitidos pelos municípios de pequeno porte para serem repassados posteriormente ao Ministério da Saúde.

A seleção de um perfil é obrigatória durante a primeira execução do software SCNES e além da escolha do perfil pode-se criar um usuário e uma senha de acesso e existe um formulário no qual deve ser preenchido com os dados fidedignos do usuário que utilizará o programa, pois posteriormente serão validados. Sendo assim

¹¹ <http://cnes.datasus.gov.br/pages/downloads/aplicativos.jsp>

um cidadão comum e idôneo jamais poderia ter acesso oficial ao software uma vez que não existe um perfil específico criado para ele, pois não é de interesse do DATASUS que ele utilize.

As principais operações possíveis de execução no software SCNES são inclusão, alteração, exclusão e transmissão de informações sobre os estabelecimentos de saúde ao Ministério da saúde. O software possui um banco de dados local e off-line, porém ele é sincronizado mensalmente com o banco de dados federal do DATASUS onde são armazenados os dados desses estabelecimentos a nível Brasil através de atualizações via internet. As rotinas diárias e forma de utilização dos usuários deste software baseiam-se em legislações publicadas no Diário oficial da União. São portarias, notas técnicas, leis das quais os usuários precisam seguir e se adequar. Uma dessas regras definidas é a obrigatoriedade da transmissão de dados de sua base local para os servidores do Ministério da Saúde mensalmente (BRASIL. Portaria nº 682, Diário Oficial da União, 2010). Dessa forma, todas as alterações que são feitas na base de dados local são repassadas para o servidor do DATASUS para validação e aprovação e posterior publicação no portal de divulgação dos dados dos estabelecimentos à sociedade. Segundo disponibilizado por DATASUS (2015a) na seção sobre limites de competências, definiu-se um cronograma que todos os municípios devem cumprir em relação à transmissão de suas bases de dados locais para os servidores do Ministério da saúde. Analisando-se o cronograma vemos que os municípios podem transmitir seus dados ao Ministério da Saúde num determinado período dentro do mês e este período não é padrão pois varia todo mês. Dentro deste período os gestores podem transmitir seus dados locais uma vez por dia. Ao realizar o envio dos dados, o servidor recebe o arquivo, armazena, e durante a madrugada ele faz o processamento do arquivo recebido, caso não existam inconsistências, no dia seguinte os dados já ficam disponíveis para visualização no site de publicação das informações sobre os estabelecimentos. Porém, pode ocorrer o carregamento parcial das informações alteradas, pois durante este período do mês aberto para transmissões, cada gestor pode transmitir dados todos os dias fazendo várias alterações, sendo assim esse é um período transitório e incerto em relação a veracidade das informações sobre os estabelecimentos. Voltando a analisar o cronograma, percebe-se que existe um período dentro de cada mês onde não é permitido o envio de dados aos servidores, pois neste curto espaço de tempo os servidores do Ministério da Saúde, processam e fazem a consistência de todas as

informações recebidas. É durante esse período que se tem as informações oficiais e definitivas sobre estes estabelecimentos, pois nesse período o servidor não aceita requisições de envio de novos dados.

3.5 ANDROID

Android é o sistema operacional open source que foi desenvolvido com a participação da empresa Google. Nesta seção do trabalho serão apresentados detalhes deste sistema. Primeiramente será apresentada um pouco de sua história de forma um pouco mais detalhada e na sequência serão mostradas as características de sua arquitetura.

3.5.1 Histórico do Android

A base do sistema operacional Android surgiu em 2003, na cidade de Palo Alto na Califórnia. Ele foi desenvolvido por Andy Rubin, Rich Miner, Nick Sears e Chris White. Estes empresários já iniciados no ramo da tecnologia, foram os fundadores da Android, Inc. A ideia principal desta empresa era a criação de dispositivos móveis mais inteligentes e que estivessem mais cientes das preferências e da localização dos seus usuário. No início a empresa desenvolvia todos os seus projetos de forma secreta. Em 2005 a Google adquiriu a empresa Android, Inc, nascendo assim a sua nova divisão de pesquisa em tecnologia móvel. Na mesma época os sistemas que dominavam o mercado de mobile eram o *Windows Mobile*, da Microsoft, e o iOS, da Apple e estas não se viram ameaçadas com os planos do Google e seus protótipos planejados juntamente com seus contratos de parcerias com algumas empresas da área de hardware. Foi somente a partir de 2007 que um grande passo foi dado e que marcou sua história fazendo outras empresas da mesma divisão de mercado começassem a se preocupar com suas posições no mercado, os grandes fabricantes como Samsung, Sony, HTC, operadoras como as americanas Sprint Nextel e T-Mobile, e fabricantes de hardware como Qualcomm e a Texas Instrumentes, além do próprio Google, reuniram-se em um consórcio de tecnologia e fundaram a OHA¹² (MEYER, 2016). Várias outras grandes empresas do ramo de telefonia, hardware, semicondutores e

¹² Open Handset Alliance - <http://www.openhandsetalliance.com>

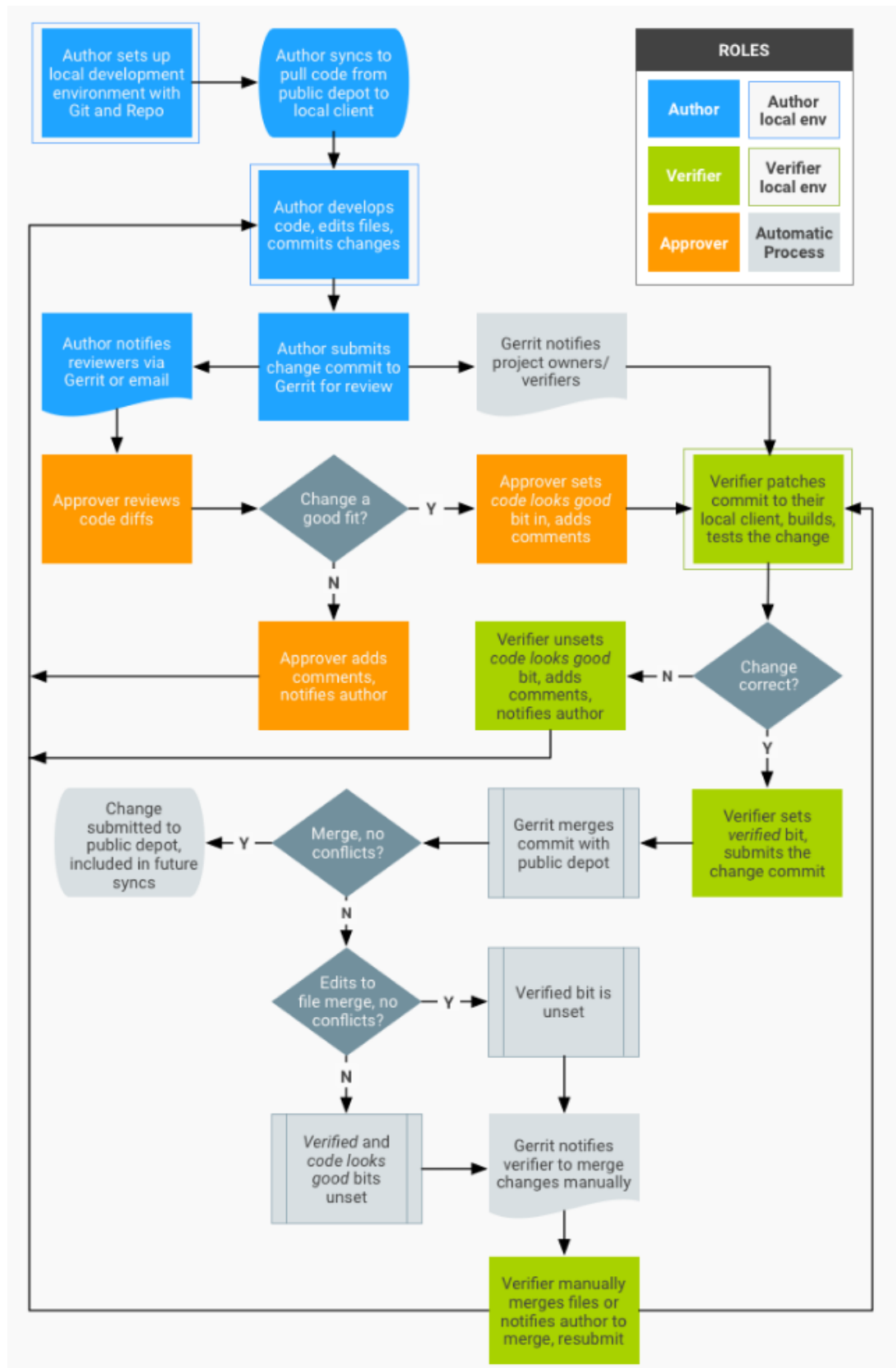
de marketing e comércio participam e contribuem para a sua evolução. Todas essas empresas compartilham suas expectativas e *insights* nos congressos promovidos pela OHA(Open Handset Alliance). O principal objetivo deste consórcio é a busca de um padrão de dispositivo móvel aberto e que atenda as necessidades atuais de todos. Uma vez que os membros pertencem aos diversos setores que envolvem a concepção de um smartphone elas possuem conhecimento suficiente para saber o que é tendência no mercado e quais são as necessidades dos consumidores. Como descrito na seção sobre a filosofia de governança em ANDROID OPEN SOURCE PROJECT (2016), ao juntar todas as experiências dessas empresas pode-se manter o Android de uma forma que seja interessante tanto para os usuários finais como para as empresas que vão utilizá-los em seus dispositivos para comercializá-los. De uma certa forma esse consórcio é o responsável por ditar as regras de como o sistema operacional irá se comportar para atender as necessidades dos consumidores finais.

Um dos principais objetivos da Google ao participar na concepção do Android foi a criação de uma plataforma padrão na qual fosse utilizada por todos os dispositivos móveis. Através dessa iniciativa ela tinha a intenção de ganhar uma grande fatia no mercado de dispositivos. Para tal ela projetou uma plataforma cheia de atrativos com o objetivo de chamar a atenção tanto da comunidade de desenvolvedores quanto das grandes empresas fabricantes de dispositivos móveis demonstrando as infinitas possibilidades benéficas adquiridas na criação de dispositivos móveis compatíveis com sua plataforma. Quando se fala em compatibilidade entende-se a compatibilidade entre o hardware desenvolvido pela empresa interessada em lançar um novo modelo de dispositivo móvel. É para essa intenção do fabricante que o projeto da plataforma Android desenvolveu um sistema de validação de compatibilidade que pode ser visto na seção sobre compatibilidade do Android em ANDROID OPEN SOURCE PROJECT (2016). Podem-se citar algumas das principais características benéficas na adoção da plataforma Android como:

- A Linguagem usada na programação de aplicativos para a plataforma é o JAVA. Muito popular, bem difundida e bem documentada na atualidade o JAVA permitiu uma rápida adaptação e uma grande aceitação pelos desenvolvedores;
- A Plataforma Android é gratuita. Para os desenvolvedores que adotam esta plataforma para desenvolver suas aplicações os custos para tal são irrisórios;

- O código fonte da plataforma é livre e qualquer indivíduo pode adquirí-lo no portal do projeto oficial da plataforma na internet. Esse fato chama bastante a atenção das grandes fabricantes de *smatphones* e *tablets* pois os mesmos podem disponibilizar o Android em seus dispositivos vendidos sem a necessidade de aquisição de licença reduzindo assim os custos de sua produção, desde que sejam aprovados pelo programa de compatibilidade de dispositivos para Android;
- Possibilidade de um alto nível de personalização da plataforma. Tal característica permite que a plataforma possa ser adaptada e ajustada de acordo com o projeto no qual será inserida. Isto é um fato que pode atrair as grandes empresas pois elas podem tematizar e modificar a plataforma para se ajustar aos seus dispositivos físicos projetados.

Como dito, a concepção do sistema operacional Android não foi de exclusividade da empresa Google. Ela é a responsável pela gestão da plataforma, porém ela possui sua licença aberta e qualquer pessoa interessada pode participar para sua melhoria. Para isto basta o interessado procurar interagir com o AOSP (*The Android Open Source Project*) e seus subprojetos que somados garantem a evolução da plataforma. Como pode ser visto na seção de regras do projeto em ANDROID OPEN SOURCE PROJECT (2016), há uma série de regras e passos que podem ser acompanhados na Figura 3, que os interessados no projeto precisam adequar-se para poder contribuir.



Fonte: ANDROID OPEN SOURCE PROJECT (2016)

A primeira versão do Android é de outubro de 2008. Como dito por Mercado

(2016), o primeiro smartphone disponibilizado comercialmente com esta versão foi o G1 fabricado pela empresa HTC. A partir de então, várias empresas começaram a adotar o Android como sistema padrão e sua popularização foi acontecendo e as empresas começaram a acompanhar e atualizar seus aparelhos lançados com as últimas versões oficiais lançadas. O Quadro 1 abaixo mostra o histórico completo de todas as versões lançadas até o presente momento da finalização deste projeto:

Quadro 1 - Plataforma de nomes de versões e níveis de API

Code name	Version	API level
Nougat	7.0	API level 24
Marshmallow	6.0	API level 23
Lollipop	5.1	API level 22
Lollipop	5.0	API level 21
KitKat	4.4 - 4.4.4	API level 19
Jelly Bean	4.3.x	API level 18
Jelly Bean	4.2.x	API level 17
Jelly Bean	4.1.x	API level 16
Ice Cream Sandwich	4.0.3 - 4.0.4	API level 15, NDK 8
Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14, NDK 7
Honeycomb	3.2.x	API level 13
Honeycomb	3.1	API level 12, NDK 6
Honeycomb	3.0	API level 11
Gingerbread	2.3.3 - 2.3.7	API level 10
Gingerbread	2.3 - 2.3.2	API level 9, NDK 5
Froyo	2.2.x	API level 8, NDK 4
Eclair	2.1	API level 7, NDK 3
Eclair	2.0.1	API level 6
Eclair	2.0	API level 5
Donut	1.6	API level 4, NDK 2
Cupcake	1.5	API level 3, NDK 1
(no code name)	1.1	API level 2
(no code name)	1.0	API level 1

Fonte: ANDROID (2016b)

Durante a linha evolutiva das versões do Android, houve um fator que ramificou tal linha durante o desenvolvimento da versão 2.3, tal fato foi o surgimento dos *tablets*. Estes possuíam telas com resoluções e poder de processamento maiores do que os smartphones cujas versões do Android até então produzidas. Devido às grandes telas, os aplicativos produzidos para as versões 2.3 do Android para smartphones não se adaptavam muito bem aos *tablets*, ficando com sua interface gráfica deformada e diferente daquelas que haviam sido projetadas. Sendo assim para resolução destas questões, as versões da plataforma 2.3.1 e 2.3.2 em diante eram

destinadas a smartphone e as versões 3.0, 3.1 e 3.2 foram adaptadas, focando dispositivos como *tablets* que possuíam telas e resoluções maiores (DEITEL; DEITEL; WALD, 2016, p. 8).

Esta separação das plataformas, uma para *tablet* e uma para smartphone, fez com que houvesse uma preocupação quanto aos frameworks usados pelos desenvolvedores durante sua criação de aplicativos. Já que houve então uma separação os desenvolvedores deveriam se preocupar agora em ter de utilizar mais ferramentas para poderem desenvolver um mesmo aplicativo, deste modo teriam o trabalho dobrado. Pensando em solucionar esta questão a Google tomou a iniciativa de modificar a plataforma de uma maneira que se pudesse unificar novamente estas linhas divisórias paralelas no desenvolvimento de aplicativos. Para tal foi desenvolvido a versão do Android 4.0 cuja proposta era a criação de uma API unificada que pudesse ser utilizada tanto para criar aplicações para *smartphones* quanto para *tablets* (SCHMIDT, 2016).

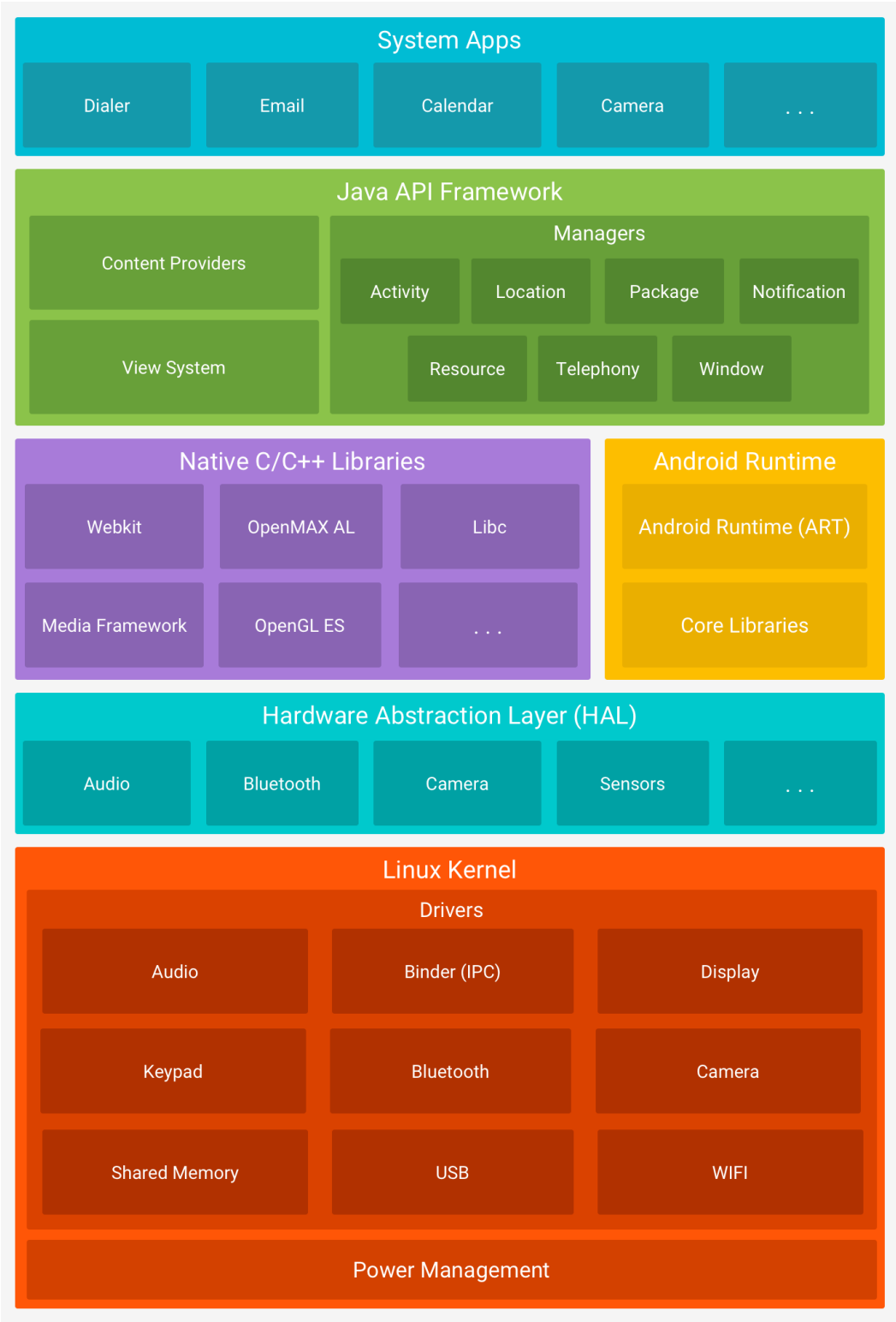
Ao longo das atualizações e melhoramentos na plataforma, novas evoluções tecnológicas foram surgindo no mercado como *SmartTvs*, sistemas operativos de multimídia para veículos automotivos e dispositivos vestíveis, como relógios e com elas a possibilidade de introduzirem a plataforma Android nesse novo universo. Dessa forma, a partir da versão 5.0 a possibilidade de desenvolvimento de aplicativos para essa nova gama de aparelhos tornou-se possível (ALVES, 2016).

3.5.2 Arquitetura do Android

Segundo pode ser visto em ANDROID (2016b), a arquitetura da plataforma possui em seu nível mais baixo e mais perto do hardware, que inclusive é quem gerencia o mesmo, o Linux *kernel*. É ele quem está por trás do funcionamento do Android. Esse *Kernel* fornece os drives para a comunicação com o hardware, como por exemplo os drivers para comunicação com a câmera, com o teclado e com o autofalante. Ele também realiza o gerenciamento de processos, Threads e de memória. O Android é uma plataforma que executa sobre o sistema operacional Linux. Existem muitas praticidades na utilização de um sistema operacional *open source* como o Linux, uma delas é o fato de sua gratuidade, outras são questões como o a praticidade de já se ter o gerenciamento de memória, processos, threads, segurança implementados e gerenciados. Dessa forma tira-se proveito inclusive da questão da

possibilidade de execução em multitarefa. A Figura 4 abaixo exibe todas as peças da arquitetura:

Figura 4 - Arquitetura da plataforma Android



Fonte: ANDROID (2016b)

Acima e em conjunto com o *kernel* do Linux existe a HAL¹³(*Hardware Abstraction Layer*). Esta camada é composta por vários módulos e interfaces que fornecem a capacidade de interação com uma linguagem de alto nível como o Java.

Logo acima da camada HAL, existem as bibliotecas que são nativas da plataforma Android e que são escritas normalmente na linguagem C ou C++. São elas que se comunicam com a camada HAL. Essas bibliotecas geralmente proveem funcionalidades como o suporte ao *SQLite*, ao motor de renderização de imagens 2D e imagens 3D, renderização de páginas WEB, entre outros.

Ao lado existe na estrutura da plataforma o *Android Runtime* que é composto pelas *Core Libraries* juntamente com a máquina virtual do Android. As *Core Libraries* são as responsáveis por disponibilizar ao desenvolvedor tudo que é utilizável pelo Java SE. A ART (*Android Runtime*) é a máquina virtual do Android utilizada para a execução dos aplicativos. Como os aplicativos são desenvolvidos e compilados em Java eles precisam de uma máquina virtual para poderem ser interpretados.

Até a versão 4.4 do Android uma outra máquina virtual era utilizada pelo Android, a *Dalvik*, esta era a máquina virtual padrão, porém o Android fornecia ao usuário a possibilidade de experimentar a ART que nesta versão era opcional (HAMANN, 2016). A partir da versão 5.0 a máquina virtual padrão tornou-se a ART.

Esta máquina virtual ART foi desenvolvida tendo como propósito melhorar a performance dos aplicativos na plataforma Android. Ela é capaz de executar os aplicativos mais rapidamente, pois em vez de trabalhar com o sistema “*Just-in-Time*” do *Dalvik*, é realizada uma compilação a frente do tempo. Os códigos são pré-compilados na linguagem de execução já durante a instalação dos aplicativos. Isso significa que a instalação dos softwares pode ser mais demorada, mas também representa mais rapidez na sua execução justamente pelo fato de a virtualização já ter acontecido e ter sido gravada na memória do dispositivo.

Diferentemente da máquina utilizada anteriormente, a *Dalvik*, a ART foi criada especificamente para o Android e para esse tipo de contexto.

A máquina Virtual do atual do Android não executa arquivos em formato *bytecode* com a extensão *.class*, assim como a JVM(Máquina Virtual do Java) na plataforma Java SE o faz, mas sim um arquivo em um formato *.dex* (*Dalvik Executable*). Na plataforma Android, o compilador, além de gerar um arquivo em

¹³ [https://source.android.com/devices/index.html#Hardware Abstraction Layer](https://source.android.com/devices/index.html#Hardware%20Abstraction%20Layer)

formato de *bytecode* e extensão *.class*, executa um passo a mais utilizando como base esse arquivo para compilá-lo em um novo com formato *.dex* para que ele então possa ser executado pela ART.

Acima dessas camadas existe o *framework* de aplicações do Android. Neste *framework* estão as funcionalidades que vão permitir ao desenvolvedor criar seus aplicativos. Dentre esses frameworks podemos citar o *Telephony manager* que disponibiliza a funcionalidade de realizar ligações telefônicas, *Resource manager* que permite ao desenvolvedor o acesso a memória interna do dispositivo, além do *Location manager* que permite a possibilidade a utilização do GPS do dispositivo. Esta parte da plataforma provê todo o gerenciamento do Android pois ela faz a comunicação com as APIs de mais baixo nível da plataforma. Quando um desenvolvedor cria um aplicativo para a plataforma ele utiliza esse *framework* para tal. Todos os aplicativos nativos do Android também foram criados utilizando esse *framework*, dessa forma ambos os tipos de aplicações, nativas ou terceiras, estão no mesmo nível e podem ser substituídos sem causar prejuízos à plataforma.

No topo desta estrutura temos a camada de aplicações onde ficam as aplicações do Android, tanto as nativas como o aplicativo de telefonia, de contatos, de mensagens, quanto as aplicações desenvolvidas por terceiros. A plataforma Android percebe as aplicações desenvolvidas por terceiros assim como as nativas da mesma forma. A grande flexibilidade desta plataforma já mencionada anteriormente está no fato de os aplicativos criados pelos desenvolvedores estarem nessa mesma camada das aplicações nativas. Logo é possível desenvolver um aplicativo que substitua outro aplicativo nativo com o objetivo de personalizar toda a interface gráfica do usuário.

4 ESTADO DA ARTE

Neste capítulo são apresentados dois sistemas que foram desenvolvidos focando o mesmo tema do trabalho desenvolvido. A busca pelo estado da arte em localização de estabelecimentos de saúde utilizou-se de vários repositórios de trabalhos acadêmicos na Internet. Para localizá-los mais rapidamente a ferramenta Google Acadêmico¹⁴ foi utilizada. Os resultados obtidos tinham baixa compatibilidade principalmente pelo fato dos sistemas propostos ou desenvolvidos relatados nos trabalhos serem totalmente dependentes de uma conexão com a internet. Um sistema selecionado por possuir características próximas aos do atual será descrito na sequência. Ele foi selecionado principalmente pelo fato de se utilizar de um aplicativo como ferramenta de busca.

4.1 SISTEMA MÓVEL DE GEOLOCALIZAÇÃO E GEOPROCESSAMENTO PARA LOCAIS DE ATENDIMENTO EM SAÚDE

O *Journal of Health Informatics (J. Health Inform.)* é uma publicação oficial da Sociedade Brasileira de Informática em Saúde. Na sua terceira edição do ano de 2014 os autores Sarrinho e Campos (2014) relataram sua experiência no desenvolvimento do projeto SAMob (Sistema Móvel de Geolocalização e Geoprocessamento para Locais de Atendimento em Saúde). O projeto SAMob tinha o objetivo de desenvolver um sistema de atendimento mobile capaz de identificar, compartilhar e qualificar centros de atendimento de saúde próximos à geolocalização atual de seus respectivos usuários via dispositivo móvel.

Com o desenvolvimento do sistema os autores esperavam fornecer aos seus usuários uma ferramenta para auxiliar na busca por locais que prestam atendimento à saúde não só em caráter emergencial, mas também local que prestam atendimento eletivo, dessa forma, aproveitando a mobilidade e praticidade dos dispositivos móveis. Os autores utilizaram duas plataformas para desenvolver o sistema: a plataforma mobile com sistema operacional Android na qual foi desenvolvido um cliente para que os seus usuários pudessem aproveitar as funcionalidades nativas dos dispositivos móveis atuais como navegabilidade GPS, onde seria possível realizar a busca e a

¹⁴ <https://scholar.google.com.br/>

qualificação dos locais de atendimento e a plataforma *web* utilizando a *Google App Engine* na qual foi desenvolvido uma aplicação servidora e administradora que disponibilizaria a sincronização de dados com os cliente *mobile*. As funcionalidades da aplicação cliente foram definidas como: obter a qualificação do local de atendimento; qualificar o local de atendimento, possibilitando comentar ou tirar fotos do local onde se encontra; obter a melhor rota para chegar ao local de atendimento utilizando a visualização em mapa; obter dados do local de atendimento; buscar por geolocalização e parâmetros de filtragem. As funcionalidades da aplicação servidora seriam: manutenção de usuários; busca por usuários; manutenção dos locais de atendimento; busca por local de atendimento.

O artigo desenvolvido pelos autores apresenta algumas características da modelagem do sistema desenvolvido através de diagramas UML. São apresentados de forma resumida os diagramas de casos de uso, de classes e de sequência.

Com a conclusão do desenvolvimento do sistema, os autores obtiveram duas aplicações rápidas, simples e interativas. Para avaliar a usabilidade do sistema desenvolvido, foi feita uma pesquisa com 15 alunos do Universidade Estadual de Feira de Santana - UEFS, para tal o sistema foi apresentado aos entrevistados que também tiveram de responder um questionário de 30 questões referentes à usabilidade do sistema. Após o resultado do teste, os autores perceberam um alto nível de satisfação em relação aos vários quesitos analisados no teste. Ao final do artigo os autores concluem sua descrição do sistema classificando-o como: uma rede social que compartilha informações de atendimento médico-hospitalar de uma forma interativa e de baixo custo. Estes ainda acrescentam na conclusão que houve sucesso no projeto pois todas as funcionalidades previstas funcionam da forma que foram planejadas.

Como trabalhos futuros os autores sugerem: a adaptação da aplicação cliente para outras plataformas móveis *iPhone*, *Windows Phone* e *Firefox OS*; a adaptação da aplicação cliente para possibilitar a inclusão de sugestão de melhorias; integração e possibilidade de compartilhamento das opiniões sobre os estabelecimentos nas redes sociais e a integração do sistema com os sistemas corporativos de saúde da rede pública. Em relação a esta última sugestão do autor para trabalhos futuros, pode-se dizer que o aplicativo que foi desenvolvido neste projeto de conclusão de curso atende alguns quesitos, uma vez que utiliza em sua base de informação os dados oficiais retirados de um sistema corporativo de saúde pública, o CNES.

As buscas por trabalhos acadêmicos que relatavam sistemas de localização

de estabelecimentos de saúde não trouxeram grandes resultados. Para o aprofundamento das buscas por sistemas similares utilizou-se a loja de aplicativos da Google. Uma grande quantidade resultados foram obtidos, porém como são aplicativos de código fechado, não se obteve muitas informações referentes a sua arquitetura interna. O aplicativo que será descrito na sequência possui muitas características parecidas com o que está sendo desenvolvido pelo autor deste trabalho, principalmente em relação a suas funcionalidades. Devido a isto ele foi o selecionado dentre os vários encontrados.

4.2 MEDSQUARE

MedSquare, segundo Medsquare (2015), é um aplicativo desenvolvido pela empresa WINTIME IPC com o objetivo de melhorar a saúde e o atendimento dos pacientes nos estabelecimentos de saúde. Ele possui duas versões, uma para plataforma Android e outra para plataforma IOS. Suas funcionalidades permitem que o usuário: encontre um estabelecimento de saúde mais próximo de sua posição geográfica atual; busque estabelecimentos pelo nome; busque estabelecimentos por especialidade médica. Os usuários do aplicativo podem avaliar os estabelecimentos de saúde e podem registrar no aplicativo o tempo de espera por atendimento no estabelecimento que este se encontra. É possível ler os comentários e avaliações dos estabelecimentos e também conversar em tempo real com quem está num determinado estabelecimento. Para que todas estas funcionalidades possam ser contempladas, o aplicativo exige do dispositivo uma conexão com a internet pois o seu banco de dados fica em um servidor no qual ele se conecta para obter informações. O site institucional do aplicativo informa que os proprietários dos estabelecimentos podem solicitar seu cadastro na base de dados da aplicação mediante a um contato prévio com a empresa. Logo, o aumento do número de estabelecimentos cadastrados está diretamente ligado com o interesse dos seus proprietários em participar do processo de cadastro.

Em agosto de 2015 foi publicada na loja de aplicativos do Google a versão 2.0.2 do aplicativo *MedSquare*. Dentre o novo visual e algumas novas funcionalidades estava a forma de alimentação de seu banco de dados de estabelecimentos. Ele agora possuía informações de todos os estabelecimentos cadastrados na base federal do CNES (Cadastro Nacional dos Estabelecimentos de Saúde). Essa estratégia tornou

possível um aumento significativo na quantidade de resultados possíveis durante uma busca por algum estabelecimento.

Ao analisar os sistemas escolhidos para conhecer o estado da arte sobre o tema, percebe-se que ambos são sistemas totalmente dependentes de uma conexão com a internet para proverem suas funcionalidades. Ambos são sistemas que usam a arquitetura cliente servidor, no qual o aplicativo, que é o cliente, solicita os dados sobre os estabelecimentos ao banco de dados que está armazenado em um servidor. E, ambos, se utilizam das várias possibilidades promovidas pelas plataformas mobile para promover facilidades aos usuários, como a possibilidade de utilizar o GPS do dispositivo, por exemplo. Percebeu-se também que a ação de inclusão de novos estabelecimentos ao banco de dados é totalmente dependente do interesse do próprio estabelecimento. Caso ele não se manifeste, ele não aparecerá no banco e os usuários não saberão de sua existência. No caso do aplicativo *MedSquare*, depois de sua última atualização esta situação deixa de ocorrer pois, segundo a própria empresa divulga, agora eles possuem em seu banco de dados todas as informações dos estabelecimentos de saúde que estão na base de dados do CNES. Entretanto, não há informações se estes dados são atualizados frequentemente.

5 DESENVOLVIMENTO DO PROJETO

Neste capítulo são apresentadas as etapas que envolveram o desenvolvimento do projeto. Na primeira seção serão tratadas as questões relacionadas a ferramenta de captura de dados, nas próximas serão descritos todos os processos envolvidos no desenvolvimento do aplicativo para a plataforma Android, assim como suas dificuldades e resultados obtidos nesta etapa.

5.1 A APLICAÇÃO DE CAPTURA DE DADOS

Foi desenvolvida uma aplicação utilizando a linguagem JAVA em sua versão 8. Como Ide, utilizou-se o *Netbeans*¹⁵ em sua versão 8. Esta aplicação utilizou uma biblioteca Java *HTML Parser JSOUP* em sua versão 1.8.2. A utilização dessa biblioteca de classes permitiu importar para dentro do programa e manipular os elementos das páginas HTML retornadas nas consultas feitas as URLs do portal do CNESNet na seção de busca por estabelecimentos de saúde para que se pudesse extrair, limpar e converter os dados que foram usados na base de dados do aplicativo. A Figura 5 ilustra a página que é retornada ao requisitar ao portal uma busca.

¹⁵ <https://netbeans.org/>

Figura 5 - Página de detalhamento do estabelecimento

Ministério da Saúde
CNESNet
 Secretaria de Atenção à Saúde
 DATASUS
 Cadastro Nacional de Estabelecimentos de Saúde

Home Institucional Serviços Relatórios Consultas Documentação Fale Conosco

Estabelecimento de Saúde

Identificação
 CADASTRADO NO CNES EM: 24/7/2004 ULTIMA ATUALIZAÇÃO EM: 18/5/2015 DATA DE ATUALIZAÇÃO LOCAL: 5/5/2015

Veja onde se localiza:

Nome: HOSPITAL UNIVERSITARIO	CNES: 3157245	CNPJ: --
Nome Empresarial: UNIVERSIDADE FEDERAL DE SANTA CATARINA	CPF: --	Personalidade: JURÍDICA
Logradouro: CAMPUS UNIVERSITARIO	Número: S/N	Telefone: (48) 3721-2074
Complemento: --	Bairro: TRINDADE	UF: SC
CEP: 88040900	Município: FLORIANOPOLIS - IBGE - 420540	Gestão: DUPLA
Tipo Estabelecimento: HOSPITAL GERAL	Sub Tipo Estabelecimento: --	Esfera Administrativa: FEDERAL
Natureza da Organização: ADMINISTRACAO INDIRETA - AUTARQUIAS	Dependência: MANTIDA	Data Expedição: 14/10/2002
Número Alvará: 3756/025 SERIE PDVS	Órgão Expedidor: SMS	

Módulos:

Básico	Conjunto	Ambulatorial	Hospitalar	Mantenedora	Profissionais
Habilitações	Regras Contratuais	Contrato de Gestão	Incentivos	Equipes	Residência Terapêutica
Telessaúde	Org. Parceiras	Ger/Adm (Terceiro)			

Fonte: DATASUS (2015b)

Após a conclusão do desenvolvimento desta ferramenta foram feitas as extrações e a montagem da base. A tecnologia da base de dados usada foi o *SQLite*.

SQLite é um banco autocontido, compacto, com suporte nativo no Android e sem necessidade de configuração ou instalação. Isto torna-o a escolha natural para um ambiente em que devemos prezar por desempenho, disponibilidade de memória e praticidade de uso (LUZZI, 2015).

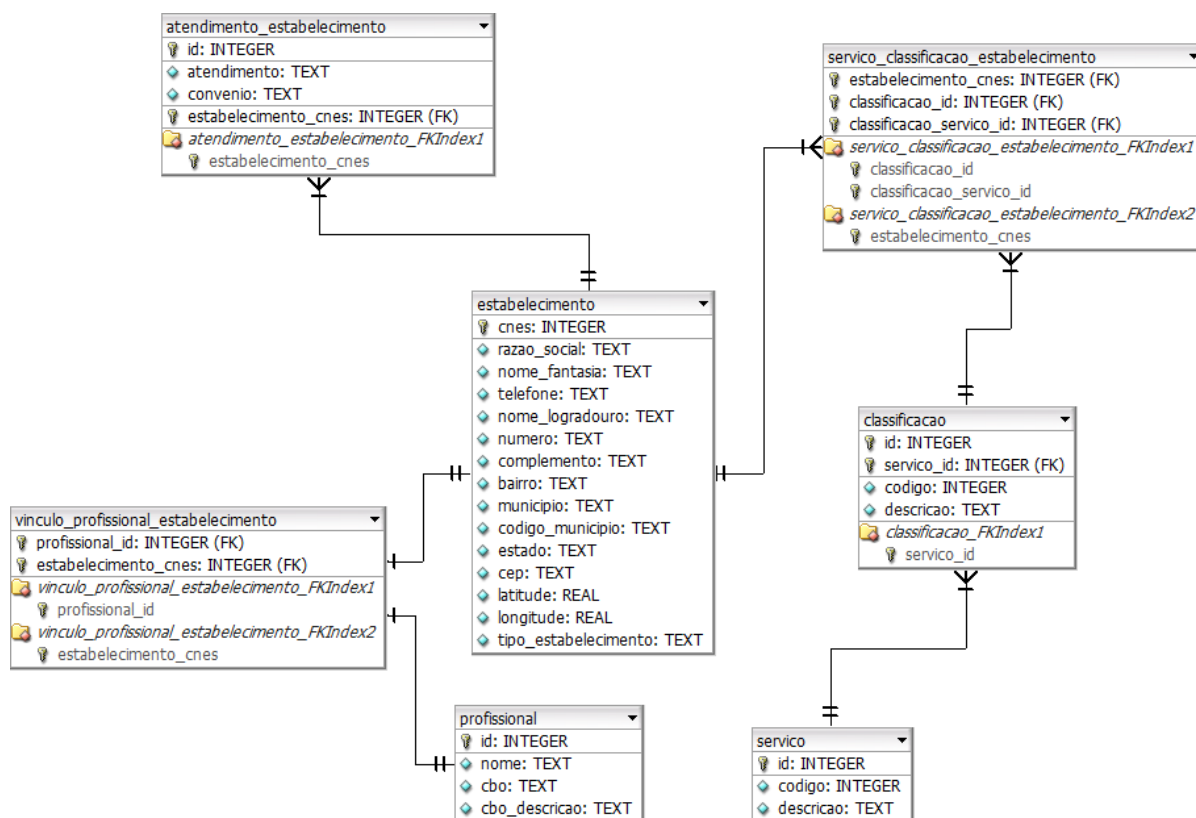
Foi utilizada a ferramenta *DB Designer Fork*¹⁶ para auxiliar no processo de construção do banco de dados. Segundo Oliveira (2008), esta ferramenta *open source* é uma versão melhorada da ferramenta *fabFORCE DBDesigner 4*¹⁷, além de ser simples, prática e fácil de usar ela permite a configuração de uma conexão direta com

¹⁶ <http://sourceforge.net/projects/dbdesigner-fork>

¹⁷ <http://www.fabforce.net/dbdesigner4>

os sistemas gerenciadores de banco de dados: *MySQL*, *SQLite*, *Oracle* e *SQL Server*. A ferramenta permite a modelagem de forma visual das tabelas e relacionamentos além da geração de scripts de criação do banco de dados na linguagem SQL para diversos gerenciadores de banco de dados. A Figura 6 apresenta a modelagem das tabelas do banco de dados *SQLite*.

Figura 6 - Modelagem do banco de dados



Fonte: Elaborada pelo autor

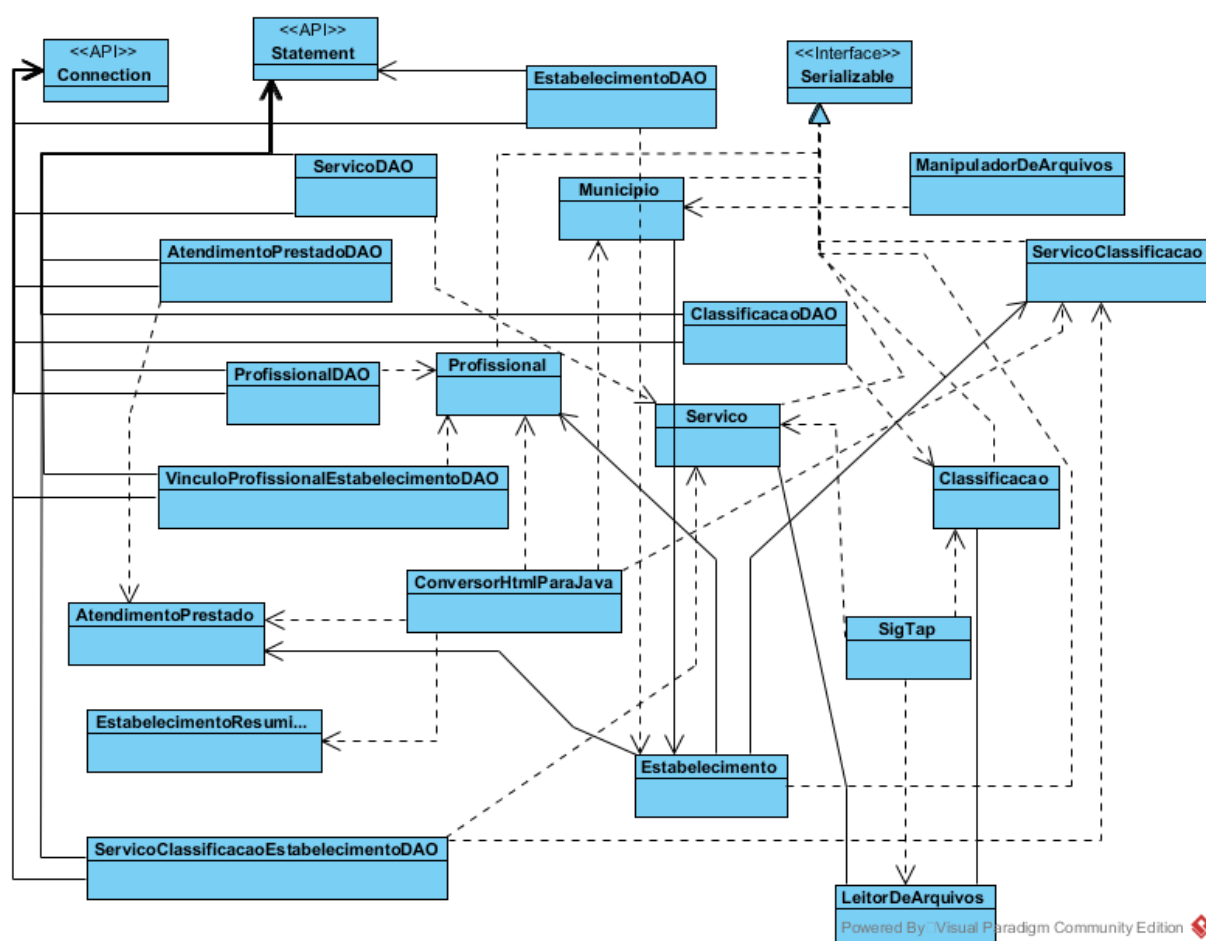
A tabela *estabelecimento* armazena os dados sobre os estabelecimentos de saúde assim como suas coordenadas geográficas. A tabela *profissional* armazena os dados dos profissionais que trabalham nos estabelecimentos, os dados armazenados são: o nome do profissional; o Código Brasileiro de Ocupação (CBO) e sua descrição. A tabela *vinculo_profissional_estabelecimento* registra quais profissionais trabalham em um estabelecimento formando um relacionamento muitos para muitos entre estas duas tabelas. A tabela *servico_classificacao_estabelecimento* armazena os serviços oferecidos pelos estabelecimentos, ela cria um relacionamento muitos para muitos entre as tabelas *estabelecimento* e *classificacao*. As tabelas *servico* e *classificacao* armazenam os dados oficiais de todos os serviços e classificações cadastrados no

SIGTAP¹⁸(Sistema de Gerenciamento da Tabela de Procedimentos do SUS). Estas informações são disponibilizadas mensalmente e de forma atualizada ao cidadão pelo portal DATASUS do Ministério da Saúde através de arquivos no formato de texto.

Com a IDE *Netbeans* foi possível incluir a biblioteca *sqlite-jdbc*¹⁹ no projeto para que as instâncias das classes responsáveis pela criação e conexão com bancos de dados *SQLite* pudessem ser utilizadas, assim, pôde-se focar o desenvolvimento na aquisição de informações através da captura dos dados das páginas HTML e na importação dos dados advindos dos arquivos de texto baixados do site SIGTAP.

Para ter uma visão geral de todas as classes pensadas para a ferramenta de captura, foi desenvolvido um diagrama de classes em UML que pode ser visualizado na Figura 7.

Figura 7 - Diagrama de classes



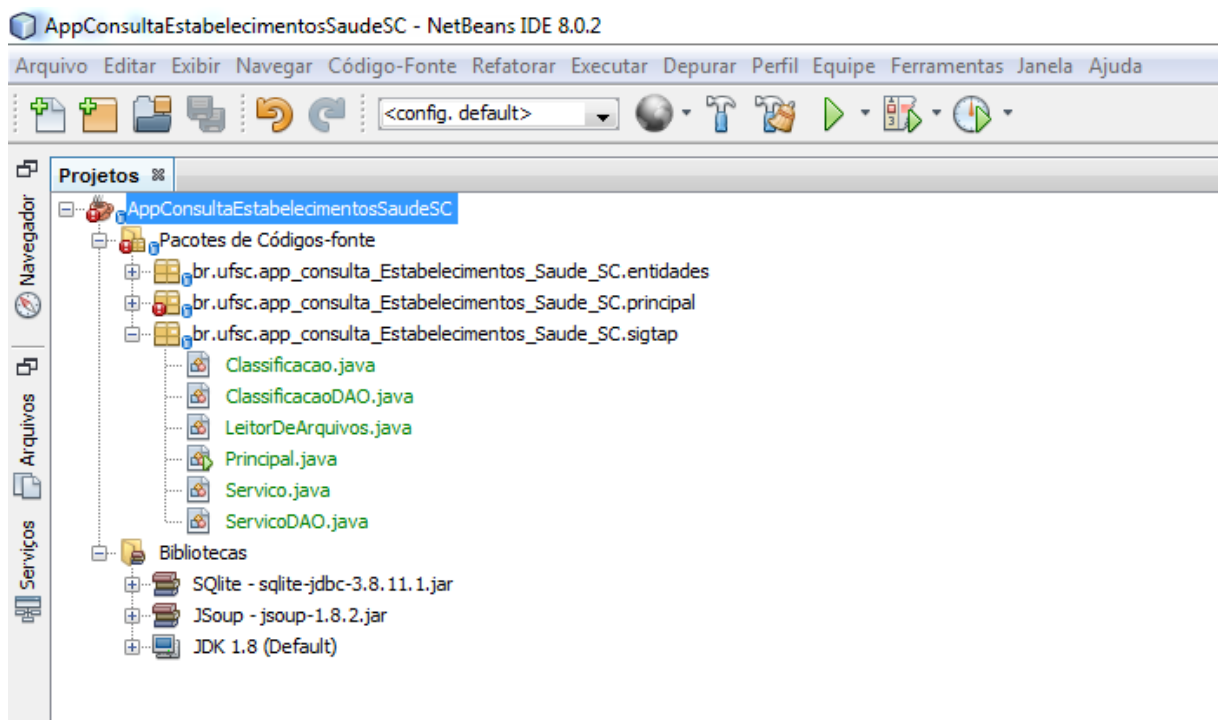
Fonte: Elaborada pelo autor

¹⁸ <http://sigtap.datasus.gov.br/tabela-unificada/app/sec/inicio.jsp>

¹⁹ <https://bitbucket.org/xerial/sqlite-jdbc>

Foram criadas algumas classes que compõem a funcionalidade de leitura de arquivos de texto desta aplicação desktop para realizar a extração dos dados contidos referentes as informações dos serviços e classificação dos SIGTAP para montagem da base de dados que será utilizada no aplicativo Android. A Figura 8 enumera as classes que servirão para este caso.

Figura 8 - Classes que fazem a leitura de arquivos de texto

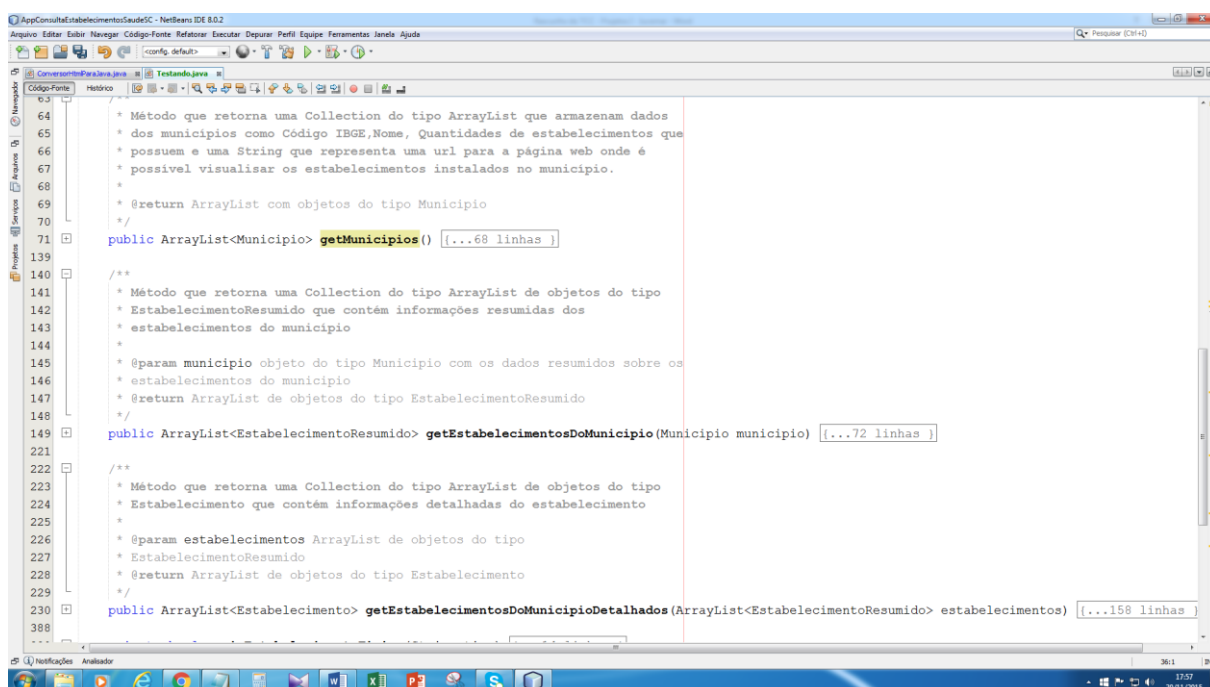


Fonte: Elaborada pelo autor

Foram criadas as classes *Servico* e *Classificacao* que são o modelo de dados que auxiliam a classe *LeitorDeArquivos* a armazenar os serviços e suas classificações em memória para posteriormente serem persistidos em banco de dados com o auxílio das respectivas classes *ServicoDAO* e *ClassificacaoDAO*, desenvolvidas seguindo o padrão *Data Access Object* (DAO).

A classe *ConversorHtmlParaJava* pode ser considerada o coração da aplicação que faz a captura dos dados. Esta classe possui as instâncias dos objetos fornecidos pela biblioteca *JSoup* que se conectam ao servidor do DATASUS e manipulam o conteúdo das páginas HTML do site. Seus principais métodos podem ser visualizados na Figura 9.

Figura 9 - Principais métodos da classe *ConversorHtmlParaJava*



Fonte: Elaborado pelo autor

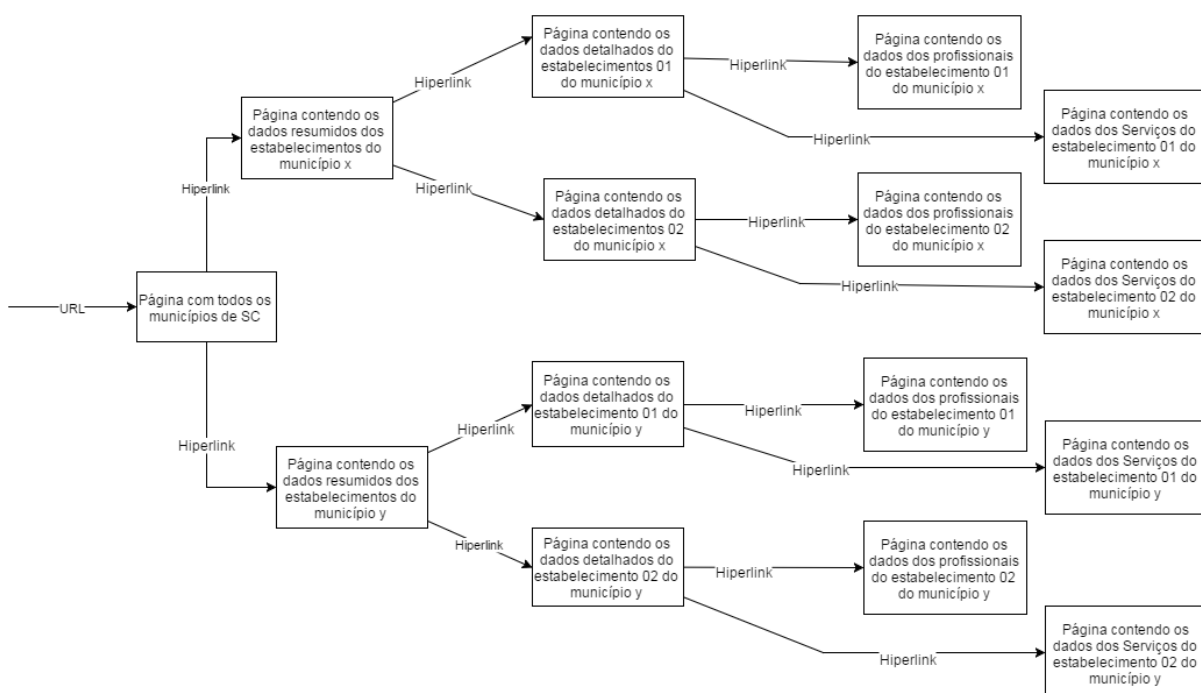
O método *getMunicipios* é o primeiro a ser executado. Ele faz uma requisição ao servidor através do endereço da página que lista todos os municípios de Santa Catarina e suas respectivas quantidades de estabelecimentos cadastrados, captura essas informações e salva em memória. Por sua vez, o método *getEstabelecimentosDoMunicipio* acessa a página que lista todos os estabelecimentos de cada município armazenando em memória dados como o nome do estabelecimento e o hiperlink que nos leva até a página detalhada de cada estabelecimento. Por fim o método *getEstabelecimentosDoMunicipioDetalhados* utiliza o *hiperlink* armazenado na etapa anterior para conectar a página que disponibiliza os dados detalhados de cada estabelecimento.

É importante ressaltar que existem vários tipos de estabelecimentos de saúde no banco de dados federal do MS. O Anexo A deste trabalho foi desenvolvido segundo (MINISTÉRIO DA SAÚDE, 2006) para descrever os tipos possíveis, porém para o escopo deste trabalho serão somente utilizados estabelecimentos dos tipos: Posto de Saúde; Centro de Saúde/Unidade Básica de Saúde; Policlínica; Hospital Geral; Hospital Especializado; Unidade Mista; Pronto Socorro Geral; Pronto Socorro Especializado; Consultório Isolado; Clínica Especializada/Ambulatório Especializado; Unidade de Serviço de Apoio de Diagnóstico e Terapia (SADT – clínicas que realizam

exames para diagnóstico); Centro de Parto Normal Isolado; Hospital / Dia- Isolado; Laboratório Central de Saúde Pública. Sendo assim o método *getEstabelecimentosDoMunicipioDetalhados* que faz uma iteração para cada estabelecimento, inicialmente analisa se o tipo do estabelecimento é compatível com o projeto, caso afirmativo ele inicia a sequência do processo de captura, caso negativo ele ignora o estabelecimento atual e passa para a próxima iteração.

Para que fosse possível conectar-se a uma página HTML utilizando as instâncias das classes disponíveis pela biblioteca *JSoup* foi importante definir um valor de parâmetro para o método *.timeout()* utilizado na chamada estática à classe que promove a conexão: *JSoup*. O parâmetro adicionado a este método define o tempo que o programa possui para tentar uma conexão bem-sucedida com o servidor de destino, caso esse tempo termine e a conexão não seja estabelecida, uma exceção é lançada informando a falha. Essa questão foi um grande desafio nesta etapa do projeto pois, precisou-se achar um valor ótimo que não fosse pequeno demais para que a exceção não chegasse a ser gerada, nem grande demais para evitar que o servidor fique com muitas requisições pendentes para atender.

Figura 10 - Árvore de links para navegação



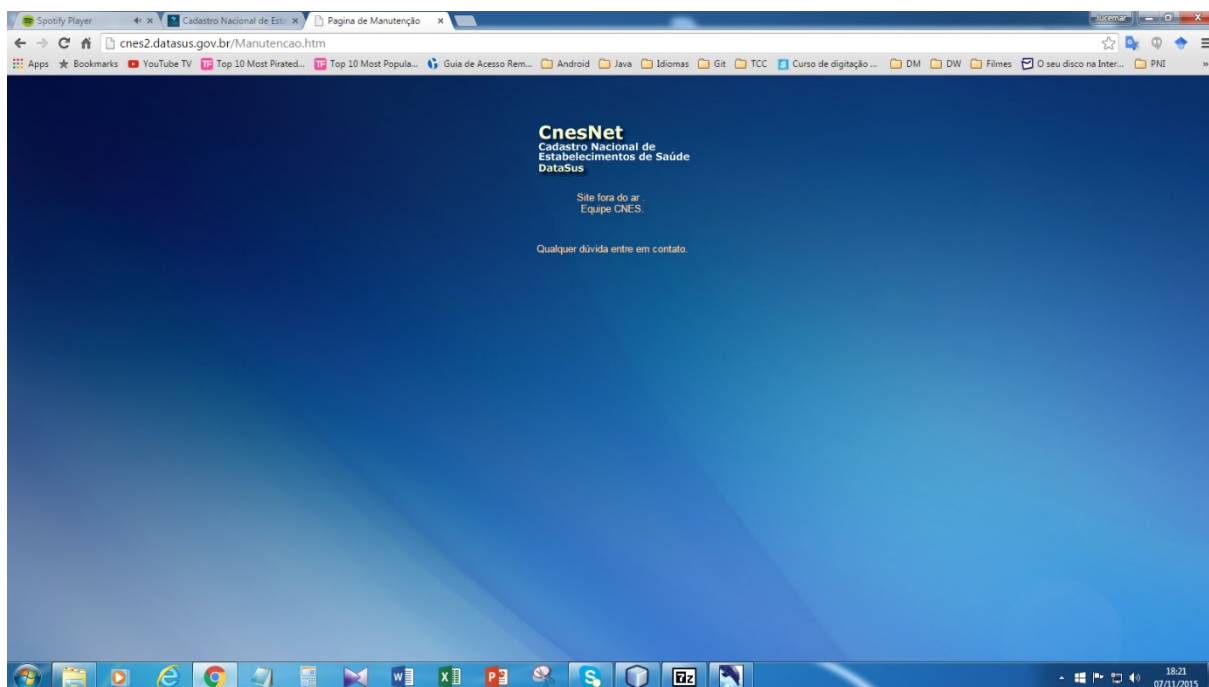
Fonte: Elaborado pelo autor

A Figura 10 demonstra a hierarquia dos links usados para navegar pelo site e

fazer as requisições ao servidor. Percebe-se claramente que a quantidade de requisições é grande. Logo, uma preocupação maior com o valor do tempo para cada requisição foi necessária para evitar que o mecanismo de segurança do servidor interprete as diversas requisições como um ataque do tipo negação de serviço (DDOS).

Várias tentativas foram feitas para encontrar um valor ótimo para o tempo de requisição, inicialmente definiu-se o valor como sendo 0. Neste contexto, ao iniciar a aplicação para realizar a captura dos dados percebeu-se que ao realizar 50% das requisições ao servidor, o mesmo começou a retornar, de forma esporádica e aleatória a página descrita na Figura 11.

Figura 11 - Site com problemas

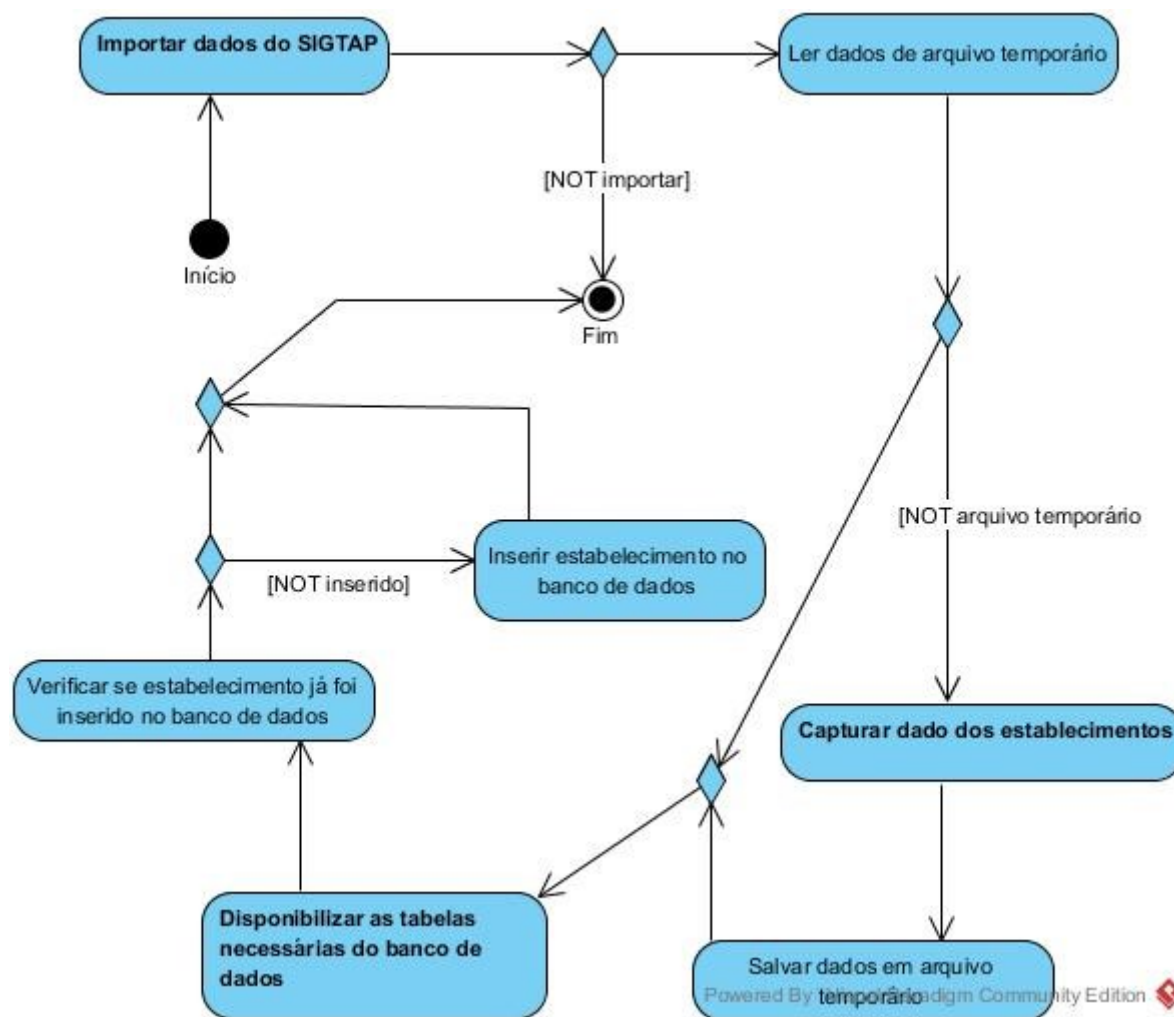


Fonte: Elaborado pelo autor

Após aumentar o valor do tempo de requisição para 200 milissegundos, a página mostrada na Figura 11 deixou de ser apresentada com muita frequência, porém, a quantidade de exceções lançadas em relação ao estouro do tempo de requisição aumentou significativamente, sendo assim, foi tratada a exceção fazendo com que se repetisse a requisição cujo tempo tenha estourado. Optou-se por aumentar o tempo de requisição para 350 milissegundos, com esse valor a quantidade de exceções geradas por estouro de tempo de requisição diminuiu.

A aplicação para Desktop desenvolvida para capturar os dados das páginas do site do CNES realiza vários procedimentos para gerar um banco de dados em formato *SQLite 3*. Para resumir todos os processos realizados pela ferramenta desenvolvida, foi criado um diagrama de atividades em UML, que pode ser visualizado na Figura 12, demonstrando o fluxo de execução da aplicação.

Figura 12 - Diagrama de atividades



Fonte: Elaborado pelo autor

Para que fosse possível adquirir as coordenadas geográficas da localização dos estabelecimentos foram utilizadas duas fontes. A primeira e prioritária foi definida como sendo a *Open Street Maps* que através da ferramenta *web open-source* desenvolvida, *nominatim*, foi possível converter o endereço comercial de cada estabelecimento em sua coordenada geográfica equivalente. Isto foi possível através de seu serviço *web* que retorna um arquivo no formato XML, contendo várias

informações sobre o endereço enviado como parâmetro, sendo uma delas, as coordenadas geográficas. Não existe um limite de requisições que podem ser feitas a essa ferramenta gratuita, contudo a precisão e o sucesso na busca por um endereço são muito baixos. A segunda fonte utilizada como sendo a secundária foi a API de mapas do Google. Essa ferramenta possui uma grande precisão nos resultados obtidos. Ela funciona de forma semelhante a primeira fonte utilizada, porém há um limite de 2.500 requisições por dia, sendo assim, criou-se um método principal *getGeocod* na classe *ConversorHtmlParaJava* que depende de outros dois métodos internos para funcionar: *getGeocodOpenStreetMaps* e *getGeocodGoogleMaps*. Inicialmente o método *getGeocod* é chamado e recebe como parâmetro o endereço do estabelecimento, esse endereço é submetido ao método *getGeocodOpenStreetMaps* e caso não retorne os dados solicitados, o método *getGeocodGoogleMaps* é acionado devido ao seu alto grau de precisão. E assim a aplicação foi adquirindo as coordenadas geográficas variando a consulta nesses dois servidores e caso nenhum retorne a resposta solicitada o método *getGeocod* retorna um valor nulo identificando assim que não conseguiu localizar coordenadas com o endereço passado como parâmetro.

Esta ferramenta desenvolvida é capaz de capturar os estabelecimentos de todos os estados do país, para tal basta alterar a URL da página inicial que contém os municípios do estado desejado, pois o site do CNES está organizado da seguinte forma: Página contendo os Estados; página contendo os Municípios e página contendo os estabelecimentos do Município. Neste projeto a ferramenta de captura foi configurada para iniciar a coleta dos estabelecimentos do Estado de SC então a URL que está dentro do programa é a que leva a páginas de municípios de SC e ao final do processo temos como resultado um arquivo de banco de dados *SQLite* com estabelecimentos de SC. Pode-se simplesmente informar a URL que leva a um outro estado como o Paraná, por exemplo e ao final do processo tem-se um arquivo de banco de dados *SQLite* com estabelecimentos do PR. Sendo assim pode-se gerar um arquivo para cada Estado, caso queira. A Figura 13 exemplifica o que foi comentado.

Figura 13 - Configuração do Estado a ser capturado

```
26
27 /**
28  * Classe responsável pela extração dos dados das páginas html do portal do
29  * CNESnet
30  *
31  * @author Jucemar
32  * @version 1.0
33  */
34 public class ConversorHtmlParaJava {
35
36     private static final String FUNCAO_PROFFISIONAIS_ESTABELECIMENTO = "Mod_Profissional.asp?VCo_Unidade=";
37     private static final String FUNCAO_SERVICO_CLASSIFICACAO = "Mod_Conj_Informacoes.asp?VCo_Unidade=";
38     private static final String FUNCAO_ATENDIMENTOS_PRESTADOS = "Mod_Bas_Atendimento.asp?VCo_Unidade=";
39     private static final String URL_PRINCIPAL = "http://cnes2.datasus.gov.br/";
40     private static final String FUNCAO_ESTABELECIMENTOS_MUNICIPIOS = "Lista_Tot_Es_Municipio.asp?Estado=42&NomeEstado=SANTA%20CATARINA";
41     private ArrayList<String> estabelecimentosFisicos;
42     int estabelecimentosValidos = 0;
43     int coordLocalizadasOSM = 0;
44     int coordLocalizadasGM = 0;
45 }
```

Fonte: Elaborado pelo autor

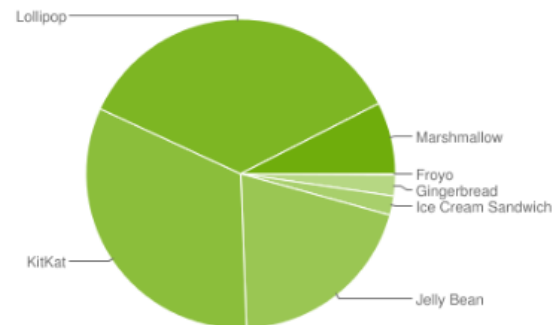
5.2 O APLICATIVO PARA DISPOSITIVOS ANDROID

A Google monitora todas as versões existentes e ativas em todos os aparelhos celulares através da sua loja de aplicativos, a *Google Play Store*. A partir do momento que um aplicativo é instalado em um dispositivo ou que uma conta de email é vinculada ao smartphone, a *Google play Store* captura a informação sobre qual a versão do Android existente naquele aparelho e contabiliza e armazena todos esses dados. Dessa forma ela pode acompanhar o crescimento e a abrangência de seu sistema nos celulares ativos hoje no mercado e divulgar ao mundo a grande fatia de mercado adquirida através de sua página de divulgação a *Dashboard*²⁰ que possui dados atualizados semanalmente como pode ser visto na Figura 14:

²⁰ <https://developer.android.com/about/dashboards/index.html#Platform>

Figura 14 - Android Dashboards - Platform Versions

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.0%
4.1.x	Jelly Bean	16	7.2%
4.2.x		17	10.0%
4.3		18	2.9%
4.4	KitKat	19	32.5%
5.0	Lollipop	21	16.2%
5.1		22	19.4%
6.0	Marshmallow	23	7.5%



Data collected during a 7-day period ending on May 2, 2016.
Any versions with less than 0.1% distribution are not shown.

Fonte: <https://developer.android.com/about/dashboards/index.html#Platform>

O desenvolvimento de um aplicativo para esta plataforma força o desenvolvedor a escolher qual a cobertura de versões do sistema da Google será focada em seu projeto. Dessa forma, quanto menor a versão focada maior será sua compatibilidade com os aparelhos ativos no mercado e maior será a sua cobertura entre os usuários. Baseando-se nesse contexto a versão escolhida para o desenvolvimento do aplicativo proposto nesse projeto foi a 4.0 (*Ice Cream Sandwich*) por ser totalmente compatível com versões superiores do Android e por atender a 97,7% de todos os *smartphones* ativos no mundo no momento da escrita deste documento.

A ideia para o desenvolvimento de um aplicativo para auxiliar na localização de estabelecimentos de saúde (clínicas médicas, clínicas especializadas, consultórios médicos especializados, clínicas que realizam exames de raios-x, exames de laboratórios) partiu da possibilidade do uso da funcionalidade de geolocalização dos atuais telefones celulares. Sendo assim, em uma situação de levantamento de requisitos o primeiro requisito funcional seria permitir ao usuário do aplicativo localizar os estabelecimentos de saúde nas proximidades. O primeiro pré-requisito não funcional poderia exigir que o dispositivo no qual o aplicativo será instalado e utilizado

deverá prover a localização geográfica (latitude e longitude) atual do usuário através do GPS e sem a necessidade de conexão com qualquer serviço de localização obtido via internet.

Para vislumbrar todas as possíveis funcionalidades do aplicativo foi desenvolvido um documento descritivo, que pode ser observado no Apêndice A. Tais requisitos foram eleitos baseando-se na ideia de facilitar a forma com que os usuários do site de divulgação dos estabelecimentos de saúde cadastrados no MS visualizam as informações. Foi pensando em uma visão sob a perspectiva de um cidadão sem conhecimentos técnicos ou avançados na área da saúde que resolveu-se reaproveitar as informações contidas nesse portal de divulgações e organizá-las de forma que pudessem auxiliar qualquer pessoa durante a sua rotina diária.

Após definidas todas as funcionalidades que iriam ser disponibilizadas pelo aplicativo no documento de requisitos criado, iniciou-se o processo de desenvolvimento por telas. Da seção 4.2.2 até a seção 4.2.7 serão apresentadas, de forma detalhada, as experiências no processo de desenvolvimento de cada uma delas. Para cada uma delas pensou-se inicialmente como ficaria a sua interface gráfica em relação ao posicionamento de botões, menus e textos e quais funcionalidades ela forneceria aos usuários. Na sequência a etapa de programação do código foi realizada. A ordem de desenvolvimento das telas segue a ordem apresentada nas seções a seguir. Esta ordem foi seguida pois cada tela desenvolvida é acessada respectivamente pela tela anterior.

O aplicativo foi desenvolvido focando-se na praticidade. Nele foram utilizados alguns dos componentes visuais mais atuais na data do desenvolvimento deste projeto. Tais componentes foram utilizados seguindo o guia de utilização do *Material Design* visto em GOOGLE DESIGN (2016), este guia aconselha como devem ser utilizados os elementos de interface gráfica dentro do aplicativo para que ele fique dentro dos novos padrões do Android.

5.2.1 O banco de dados

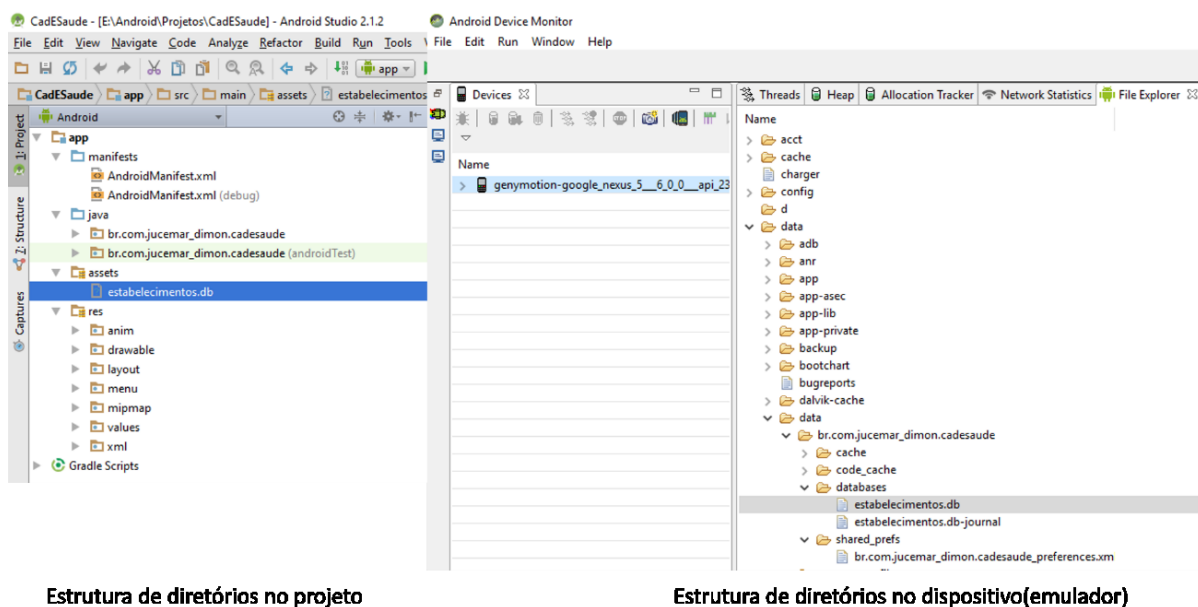
O aplicativo conta com um banco de dados interno. Trata-se de um arquivo único que é gerado na etapa de captura dos dados dos estabelecimentos usando o gerenciador de banco de dados *SQLite* em sua versão 3. Este arquivo é a base para as consultas que são feitas no aplicativo que se utiliza das soluções nativas para

manipulação de bancos e dados *SQLite*²¹ no SDK do Android. Esse arquivo de banco de dados é acessado pelas classes desenvolvidas para manipular seus dados. Na árvore de diretórios do projeto do aplicativo no software *Android Studio* existem pastas específicas para armazenamentos de arquivos de tipos específicos, segundo o guia do usuário apresentado em ANDROID STUDIO (2016), existe um diretório no projeto do aplicativo no qual pode-se armazenar um arquivo num formato qualquer que o mesmo não é alterado no processo de compilação do código fonte e criação do arquivo APK utilizado para instalação do aplicativo no dispositivo, o diretório *assets*. Este diretório mantém os arquivos intactos e possibilita a sua leitura através de *streams* de *bytes*. O arquivo contendo o banco de dados do aplicativo foi salvo neste diretório. Tal decisão foi tomada pois o aplicativo não criará um arquivo de banco de dados durante sua primeira execução, como geralmente é feito ao utilizar-se as classes para o gerenciamento de banco de dados nativos no android como *SQLiteOpenHelper*²². Neste contexto já existe um banco de dados criado e preenchido, sendo assim é necessário armazená-lo dentro do dispositivo para que se possa utilizá-lo conectando-se a ele e realizando consultas. Como afirmado, o conteúdo da pasta *assets* é armazenado dentro do arquivo no formato APK utilizado na instalação do aplicativo no dispositivo, sendo assim a técnica utilizada para transferir o banco de dados para o dispositivo foi o armazenamento e transferência para pasta final onde o aplicativo instalado fica armazenado dentro da estrutura de diretórios de arquivos da plataforma Android. A Figura 15 demonstra ao lado esquerdo a localização do arquivo de banco de dados no projeto no *Android Studio* e ao lado direito a sua localização no diretório do aplicativo instalado dentro do dispositivo.

²¹ <https://developer.android.com/reference/android/database/sqlite/package-summary.html>

²² <https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html>

Figura 15 - Estrutura de diretórios: projeto Android X Emulador



Estrutura de diretórios no projeto

Estrutura de diretórios no dispositivo(emulador)

Fonte: Elaborado pelo autor

Após a autorização da instalação do aplicativo pelo usuário, os arquivos de instalação contidos no arquivo APK de instalação são transferidos para o dispositivo e durante a primeira execução do aplicativo, o processo de armazenamento do arquivo de banco de dados em seu local de destino na pasta *data/data/br.com.jucemar_dimon.cadesaude/databases* é completado deixando o mesmo disponível para conexões. Todo o gerenciamento da conexão com o banco de dados é realizado através de uma classe nomeada de *BancoDeDados* que herda todas as características da classe nativa da plataforma Android para manipulação de conexões com bancos de dados *SQLite*, a classe *SQLiteOpenHelper*²³. Utilizando os métodos desta classe é possível criar, atualizar, abrir e fechar conexões com o banco. Para este aplicativo criou-se um método personalizado para permitir que a estratégia aqui desenvolvida fosse aplicável. A Figura 16 demonstra algumas linhas de código usadas na solução.

²³ <https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html>

Figura 16 - Classe gestora de conexões com o banco de dados

```
17 public class BancoDeDados extends SQLiteOpenHelper {
18     public static String BD_CAMINHO = "/data/data/br.com.jucemar_dimon.cadesaude/databases/";
19     public static String BD_NOME = "estabelecimentos.db";
20     public static final int BD_VERSAO = 201607;
21     private static BancoDeDados instancia;
22     private Context contexto;
23
24     public static BancoDeDados getInstancia(Context contexto) {
25         if (instancia == null) {
26             instancia = new BancoDeDados(contexto);
27             String myPath = BD_CAMINHO + BD_NOME;
28             File banco = new File(myPath);
29             if (banco.exists() && !banco.isDirectory()) {
30                 banco.delete();
31             }
32             String caminho = "/data/data/br.com.jucemar_dimon.cadesaude/databases/";
33             File pasta = new File(caminho);
34             pasta.mkdirs();
35             File bancoPronto = new File(pasta, BD_NOME);
36             try {
37                 bancoPronto.createNewFile();
38                 Log.e("BancoDeDados", "arquivo de banco de dados não existe, ele será copiado");
39                 instancia.copiarBancoDeDados();
40             } catch (IOException e) {
41                 Log.e("BancoDeDados", "Houve algum erro durante a criação ou cópia do banco de dados");
42             }
43         }
44         return instancia;
45     }
46
47     private BancoDeDados(Context contexto) {...}
48
49     @Override
50     public void onCreate(SQLiteDatabase db) {...}
51
52     public void copiarBancoDeDados() {...}
53
54 }
```

Fonte: Elaborado pelo autor

Como visto na figura acima, a classe usa o padrão de projetos *Singleton* (FREEMAN et al., 2009, p. 125) na qual somente uma instância da classe *BancoDeDados* é disponibilizada por vez. A cada solicitação de uma instância da classe é feita uma verificação em relação à existência e acessibilidade do arquivo de banco de dados. No caso da não existência do mesmo, como no contexto da primeira execução do aplicativo logo após a instalação, uma cópia dele é feita para o diretório *databases* para que a conexão possa ser criada.

5.2.2 A tela inicial

A primeira tela visualizada pelo usuário do aplicativo é tela de apresentação. Ela é muito simples pois contém apenas o logotipo, a descrição e a versão do software, porém ela tem um papel fundamental pois além de apresentar o aplicativo atraindo a atenção do usuário, permite que um processamento seja feito em segundo plano para verificar se o aplicativo já foi utilizado alguma vez ou se esta é a sua primeira

execução. Caso seja a sua primeira execução, o usuário é levado à tela de configuração inicial do aplicativo, caso contrário a tela principal é apresentada ao mesmo. A Figura 17 ilustra a tela e a Figura 18 contém um trecho do código que executa esta ação.

Figura 17 - A tela de apresentação do aplicativo



Fonte: Elaborado pelo autor

Figura 18 - Trecho de código fonte que inicia a tela de apresentação

```
35 new Handler().postDelayed(new Runnable() {
36     /*
37      * Exibindo a tela inicial.
38      */
39     @Override
40     public void run() {
41         int qtde=PreferenceManager.getDefaultSharedPreferences(getApplicationContext())
42             .getStringSet("municipios preferidos", new HashSet<String>()).size();
43         // Esse método será executado sempre que o tempo acabar
44         // E inicia a activity principal
45         if(qtde>0){
46             Intent i = new Intent(TelaInicialActivity.this, MainActivity.class);
47             startActivity(i);
48         }else{
49             Intent i = new Intent(TelaInicialActivity.this, PrimeirosPassosActivity.class);
50             startActivity(i);
51         }
52         // Fecha a tela de apresentação
53         finish();
54     }
55 }, SPLASH_TIME_OUT);
```

Fonte: Elaborado pelo autor

Segundo a documentação oficial disponibilizada pelo site *Developer Android*²⁴, um *Handler* é um mecanismo que permite enviar uma mensagem a *thread* principal do aplicativo, a *UI thread*. Somente a *thread* principal faz o processamento dos elementos da interface gráfica dos aplicativos na plataforma Android, logo, um *Handler* avisa esta *thread* de que ela precisa realizar certo processamento. É isto que está sendo feito na Figura 18. Um *Handler* executa as linhas de código dentro do método *run()*, após o tempo de 2000 milissegundos que foi definido pela constante *SPLASH_TIME_OUT*. A variável *qtde* armazena a quantidade de municípios que o usuário definiu em sua configuração. Caso ela seja menor ou igual a zero a tela de configurações é aberta com auxílio da classe *Intent*²⁵.

5.2.3 A tela de favoritos

A tela de favoritos, vista na Figura 19, possui uma área na qual os estabelecimentos anteriormente definidos como favoritos pelo usuário aparecem listados para que o mesmo tenha seu acesso rápido.

²⁴ <https://developer.android.com/reference/android/os/Handler.html>

²⁵ <https://developer.android.com/reference/android/content/Intent.html>

Figura 19 - Tela de estabelecimentos favoritos



Fonte: Elaborado pelo autor

Essa área do aplicativo possui dois botões de ação flutuante que se destacam visualmente do restante da interface gráfica justamente para chamar a atenção do usuário. Segundo a seção sobre botões de ação flutuante em GOOGLE DESIGN (2016), um botão desta categoria é o componente visual que conterá as ações principais do aplicativo ou de uma tela em questão. Seguindo esta boa prática definida pela Google, tal categoria de botão foi adicionado a esta tela do aplicativo. Foram adicionados dois botões, cada um deles com uma ação específica: um para permitir que o usuário seja direcionado para a tela de busca por estabelecimentos utilizando suas coordenadas geográficas e outro para levar o usuário à tela que permitirá ao mesmo a busca por estabelecimentos por parâmetros de interesse.

Há ainda nesta tela, como visto na Figura 20, um menu de opções padrão localizado no canto superior direito, dentro de um componente da interface gráfica

denominado *actionBar*²⁶, o qual possibilita o armazenamento e gerenciamento de componentes como menus de ação e buscas entre outros. Neste menu foram incluídas algumas ações que possibilitam ao usuário: ir à área de configurações central do aplicativo; ir até a tela de ajuda e visualizar informações gerais sobre a aplicação.

Figura 20 - Menu de configurações



Fonte: Elaborado pelo autor

No desenvolvimento destas telas das Figuras 19 e 20, utilizou-se o componente de interface gráfica *RecyclerView*²⁷. Este componente fornece uma lista que pode armazenar e exibir uma quantidade bastante grande de itens na tela da aplicação. Nesta tela, todos os estabelecimentos os quais o usuário visualizou e

²⁶ <https://developer.android.com/reference/android/app/ActionBar.html>

²⁷ <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html>

marcou como favorito são listados dentro da *RecyclerView*. Este por sua vez permite acesso direto à visualização dos dados do estabelecimento com um simples toque interativo na área onde fica localizado o título deste estabelecimento dentro de um item na lista da *RecyclerView*.

Os estabelecimentos podem ser marcados como preferidos pelo usuário durante a visualização de seus dados detalhados. O registro do estabelecimento marcado como favorito é salvo na memória interna do aplicativo com o auxílio da interface *SharedPreferences*²⁸ disponibilizada pelo SDK do Android. Esta API facilita a programação do processo de salvamento dos dados na memória do dispositivo. A Figura 21 demonstra como o processo de salvamento pode ser simplesmente feito através de poucas linhas de código. Com a execução do método descrito na figura, os dados contidos na estrutura de dados do tipo *Set* de *Strings*²⁹ são salvos em um arquivo no formato XML. A Figura 22 revela o diretório dentro da memória interna do dispositivo onde os dados são salvos caso se utilize a interface *SharedPreferences* para realizar tal ação. Para que fosse possível a visualização deste diretório fez-se necessário a utilização de uma ferramenta contida na IDE *Android Studio* onde é possível visualizar informações sobre os recursos do dispositivo que está sendo emulado num determinado momento, a ferramenta em questão chama-se *Android Device Monitor*³⁰. Tal ferramenta foi utilizada nesse contexto para visualizar a árvore de diretórios gerada pela instalação do aplicativo no sistema operacional Android para comprovar o funcionamento da estratégia utilizada para armazenar a informação sobre estabelecimentos favoritos.

²⁸ <https://developer.android.com/reference/android/content/SharedPreferences.html>

²⁹ <https://developer.android.com/reference/java/lang/String.html>

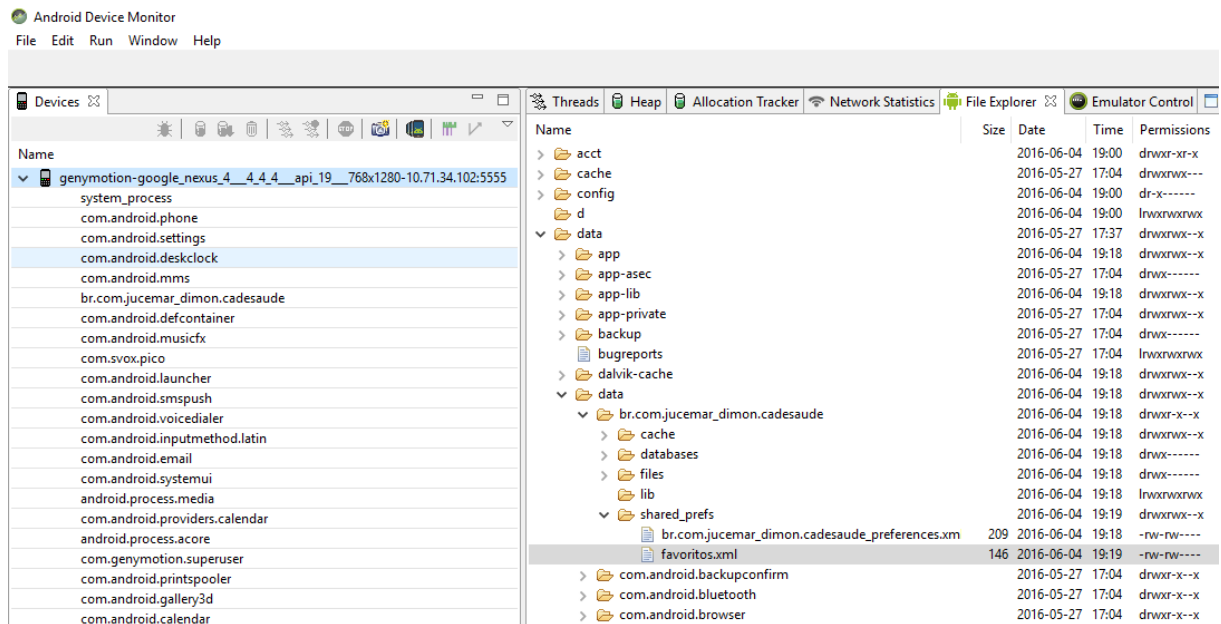
³⁰ <https://developer.android.com/studio/profile/monitor.html>

Figura 21 - Código fonte da definição do estabelecimento como favorito

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Intent;
4 import android.content.SharedPreferences;
5 import java.util.ArrayList;
6 import java.util.HashSet;
7 import java.util.List;
8 import java.util.Set;
9
10 public class EstabelecimentoActivity extends AppCompatActivity {
11     private ArrayList<ServicoClassificacao> listaDeServicos;
12     private SharedPreferences.Editor editorConfig;
13     private Estabelecimento estabelecimento;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         listaDefavoritos=new HashSet<String>();
19         estabelecimento = getIntent().getParcelableExtra("estabelecimento");
20     }
21
22     private void toggleFavorito() {
23         if(listaDefavoritos.contains(String.valueOf(estabelecimento.getCnes()))){
24             listaDefavoritos.remove(String.valueOf(estabelecimento.getCnes()));
25         }else{
26             listaDefavoritos.add(String.valueOf(estabelecimento.getCnes()));
27         }
28         editorConfig.putStringSet("listaDeFavoritos", listaDefavoritos);
29         editorConfig.apply();
30     }
31 }
```

Fonte: Elaborado pelo autor

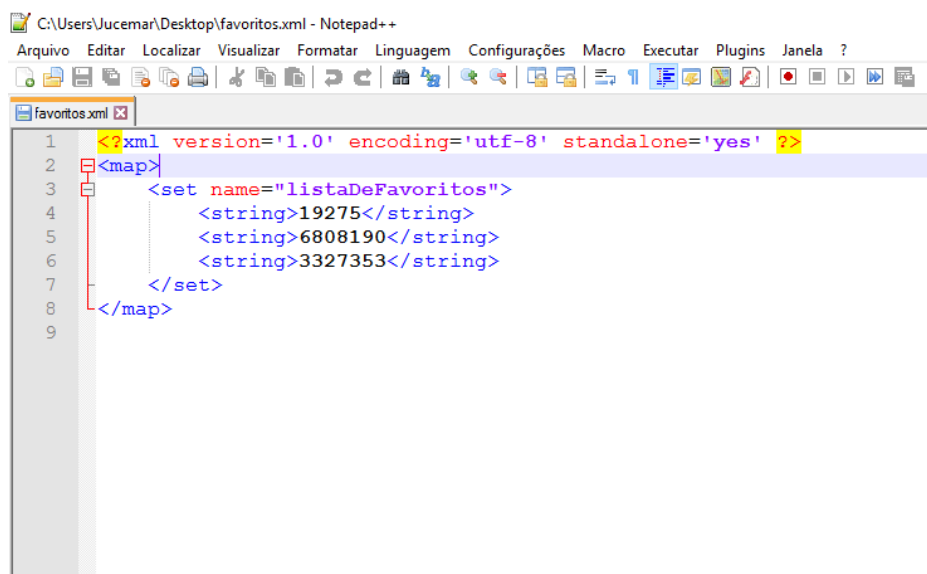
Figura 22 - Diretório no dispositivo onde é salvo o estabelecimento favorito



Fonte: Elaborado pelo autor

Para armazenar os dados sobre os favoritos marcados utilizou-se um arquivo no formato XML, como visto na Figura 22. Optou-se por armazenar o código CNES de cada estabelecimento definido como favorito para que a sua posterior recuperação no banco de dados pudesse se tornar mais eficiente. A Figura 23 mostra a estrutura interna do arquivo que mapeia os códigos CNES dos estabelecimentos definidos como favoritos.

Figura 23 - Estrutura interna do arquivo de estabelecimentos favoritos



Fonte: Elaborado pelo autor

A classe principal da aplicação que contém a tela de favoritos é chamada de *MainActivity*. Ela é uma classe que herda da classe *AppCompatActivity*³¹ que é a classe que pertence à biblioteca de suporte desenvolvida pela empresa, que permite que certas funcionalidades de versões mais recentes do Android funcionem em versões mais antigas. A classe *MainActivity* possui uma classe interna chamada de *CarregadorDeDados* que herda da classe *AsyncTask*³². A *AsyncTask* é uma classe pertencente a biblioteca nativa do SDK do Android que permite abstrair o processo de trabalhar com *multithreads* no desenvolvimento, esta classe permite configurar uma tarefa para que seja executada em um *thread* separada da *thread* principal, a UI Thread, do aplicativo e faz isso de forma assíncrona impedindo que a interface gráfica do aplicativo que é gerenciado pela thread principal gaste seu tempo de processamento com outras tarefas secundárias. A classe privada *CarregadorDeDados* executa suas tarefas em uma *thread* em segundo plano através do método *doInBackground*³³, este, após o término da execução das linhas de códigos contidas, chama o método *onPostExecute*³⁴ o qual faz a comunicação entre as ações realizadas na *thread* de segundo plano com a *thread* de primeiro plano permitindo assim ambas interagirem. O processo de carregamento dos estabelecimentos cujo usuário definiu como favorito é feito através da instanciação da classe *CarregadorDeDados* e execução do método que inicia o processo a partir da classe *MainActivity*. Na sequência, é extraído da memória interna a lista de estabelecimentos favoritos através do objeto da classe *PreferenceManager*³⁵. Em posse desta lista e de uma instância da classe *EstabelecimentoDAO*, que é a responsável por abstrair a comunicação com a tabela *estabelecimento* do banco de dados, executa-se o método *listaEstabelecimentosFavoritos* passando como parâmetro a lista, tem-se como resultado uma coleção de estabelecimentos, caso haja algum favorito definido anteriormente. Após o final desses processos o método *onPostExecute* é chamado e a *RecyclerView* tem seu adaptador preenchido com estes dados e logo ela está pronta para apresentar os estabelecimentos na tela. Na Figura 24 pode-se visualizar o código fonte mencionado.

³¹ <https://developer.android.com/reference/android/support/v7/app/AppCompatActivity.html>

³² <https://developer.android.com/reference/android/os/AsyncTask.html>

³³ [https://developer.android.com/reference/android/os/AsyncTask.html#doInBackground\(Params...\)](https://developer.android.com/reference/android/os/AsyncTask.html#doInBackground(Params...))

³⁴ [https://developer.android.com/reference/android/os/AsyncTask.html#onPostExecute\(Result\)](https://developer.android.com/reference/android/os/AsyncTask.html#onPostExecute(Result))

³⁵ <https://developer.android.com/reference/android/preference/PreferenceManager.html>

Figura 24 - método de execução em segundo plano

```

254 private class CarregadorDeDados extends AsyncTask<Void, Void, Void> {
255     private Context contexto;
256
257     public CarregadorDeDados(Context contexto) {
258         this.contexto = contexto;
259     }
260
261     @Override
262     protected Void doInBackground(Void... params) {
263         listaDeFavoritos = config.getStringSet("listaDeFavoritos", new HashSet<String>());
264         listaDeMunicipios = PreferenceManager.getDefaultSharedPreferences(contexto)
265             .getStringSet(getString(R.string.pref_municipios_preferidos_key), new HashSet<String>());
266         estabDAO = EstabelecimentoDAO.getInstance(getApplicationContext());
267         estabelecimentos = estabDAO.listaEstabelecimentosFavoritos(listaDeFavoritos);
268         //Log.e("serch", dadosDeBuscaNaTela.trim().length() > 0 ? "SearchView" + " 1" : "Nada" + " 1");
269         if (dadosDeBuscaNaTela.trim().length() > 0) {
270             bkpEstabelecimentos = new ArrayList<>();
271             bkpEstabelecimentos = estabelecimentos;
272             ArrayList<Estabelecimento> estabelecimentosTemp = new ArrayList<>();
273             for (Estabelecimento e : estabelecimentos) {
274                 if (e.getNomeFantasia().toLowerCase()
275                     .contains(dadosDeBuscaNaTela.toLowerCase())
276                     || e.getRazaoSocial().toLowerCase()
277                     .contains(dadosDeBuscaNaTela)) {
278                     estabelecimentosTemp.add(e);
279                 }
280             }
281             estabelecimentos = estabelecimentosTemp;
282         }
283         dadosDeBuscaNaTela = "";
284         return null;
285     }

```

Fonte: Elaborado pelo autor

A tela de favoritos ainda conta com a funcionalidade de busca pelo nome de algum estabelecimento que está armazenado dentro na *RecyclerView*. A lupa localizada no canto superior direito da tela ao ser tocada expande abrindo um campo de busca de texto. A base dessa funcionalidade é fundamentada através do framework de busca oficial do Android, e através da *SearchView*³⁶ foi possível adicionar tal funcionalidade ao aplicativo. Um dos passos para configuração do *framework* de *search* é a definição no arquivo *Android Manifest*³⁷ de qual será a *Activity*³⁸ que será responsável por processar as buscas. Como as buscas serão feitas na lista de estabelecimentos favoritos, a configuração foi feita nas propriedades da *Activity MainActivity* no arquivo de configurações *Android Manifest*. A Figura 25 detalha como foi definida a funcionalidade de busca nesta tela do aplicativo.

³⁶ <https://developer.android.com/reference/android/widget/SearchView.html>

³⁷ <https://developer.android.com/guide/topics/manifest/manifest-intro.html>

³⁸ <https://developer.android.com/guide/components/activities.html>

Figura 25 - Trecho de código que define a *SearchView*

```
162         @Override
163         public boolean onCreateOptionsMenu(Menu menu) {
164             MenuInflater menuInflater = getMenuInflater();
165             menuInflater.inflate(R.menu.menu_main, menu);
166             MenuItem searchItem = menu.findItem(R.id.action_search);
167             SearchManager searchManager = (SearchManager) MainActivity.this
168                 .getSystemService(Context.SEARCH_SERVICE);
169             SearchView searchView = null;
170             if (searchItem != null) {
171                 searchView = (SearchView) searchItem.getActionView();
172             }
173             if (searchView != null) {
174                 searchView.setSearchableInfo(searchManager.getSearchableInfo(MainActivity.this
175                     .getComponentName()));
176             }
177             MenuItemCompat.OnActionExpandListener expandListener = new MenuItemCompat
178                 .OnActionExpandListener() {
179                 @Override
180                 public boolean onMenuItemActionCollapse(MenuItem item) {
181                     carregaDados();
182                     return true;
183                 }
184
185                 @Override
186                 public boolean onMenuItemActionExpand(MenuItem item) {
187                     // Fazer algo quando o menu é expandido
188                     return true;
189                 }
190             };
191             MenuItemCompat.setOnActionExpandListener(searchItem, expandListener);
192             return super.onCreateOptionsMenu(menu);
193         }
```

Fonte: Elaborado pelo autor

O método *onCreateOptionsMenu*³⁹ é um método padrão de uma *Activity* que é executado pela plataforma do Android seguindo o ciclo de vida de uma *Activity*. Neste método é definido como o menu e seus itens serão apresentado na interface gráfica do aplicativo. Como a funcionalidade de busca é ativada através do toque de um item de menu, a sua programação deve ser feita dentro deste método. A *SearchView* foi definida e foi adicionado um *listener*⁴⁰ para avisar sobre o momento em que o usuário toca no botão de busca. Com isto o campo de busca expande-se para que o usuário possa digitar a palavra a ser buscada e com o aviso do *listener* aciona-se o método que realiza o procedimento de busca em segundo plano, como mencionado na Figura 24. Essa busca verifica se algum termo foi digitado no campo de busca da tela, caso afirmativo é feita uma verificação na lista de estabelecimentos adquiridos do banco de dados dos estabelecimentos que estão atualmente na lista de

³⁹ <https://developer.android.com/guide/topics/ui/menus.html>

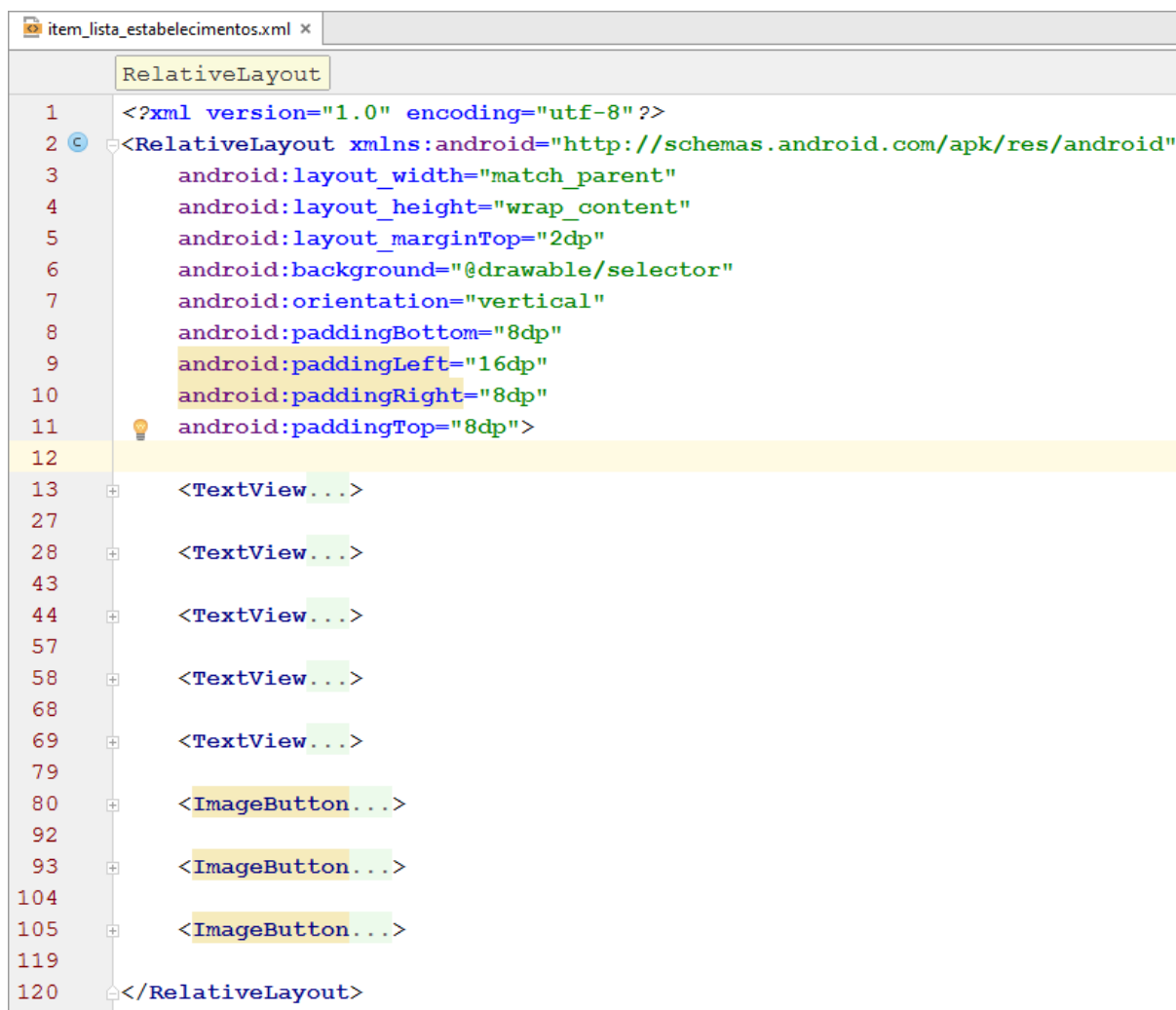
⁴⁰

[https://developer.android.com/reference/android/view/MenuItem.html#setOnActionExpandListener\(android.view.MenuItem.OnActionExpandListener\)](https://developer.android.com/reference/android/view/MenuItem.html#setOnActionExpandListener(android.view.MenuItem.OnActionExpandListener))

favoritos atual e que possuem este termo em sua razão social ou nome fantasia, caso afirmativo esses estabelecimentos são postos em uma outra lista, a qual será apresentada na tela através de uma outra *RecyclerView*.

Para organizar as informações que serão exibidas sobre cada estabelecimento de saúde que será adicionado dentro da *RecyclerView* foi criado um adaptador de dados personalizado estendendo a classe *RecyclerView.Adapter*⁴¹. Estendendo esta classe, reescrevendo os seus métodos para que os elementos de interface gráfica que foram definidos através de um arquivo XML que pode ser visualizado na Figura 26, obteve-se como resultado os elementos de interface gráfica contidos da figura 27.

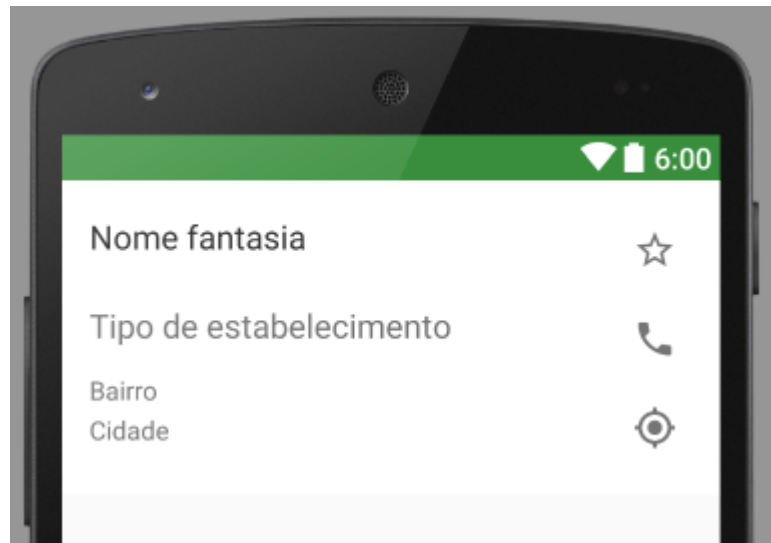
Figura 26 - Layout do modelo de apresentação dos dados dos estabelecimentos



Fonte: Elaborado pelo autor

⁴¹ <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.Adapter.html>

Figura 27 - Resultado do layout na interface gráfica



Fonte: Elaborado pelo autor

Esse modelo de lista de estabelecimentos de saúde é utilizado em todas as telas que apresentam listas no aplicativo. Ele apresenta as informações mais relevantes sobre os dados de um estabelecimento de saúde. A direita existe uma coluna com três ícones e cada um deles dispara uma função. Estes ícones são *ImageButton*⁴² que possuem métodos que são chamados por um *listener* caso haja um *click* em um deles:

- O ícone de estrela ao ser pressionado, define o estabelecimento como um dos favoritos pelo usuário;
- O ícone de telefone realiza uma chamada telefônica para o número de telefone cadastrado no estabelecimento, utilizando o aplicativo padrão de ligações do *smartphone* onde ele está instalado;
- O ícone em formato de GPS, abre o aplicativo padrão de visualização de mapas do *smartphone* e faz a marcação do local no mapa da localização deste estabelecimento utilizando as coordenadas geográficas cadastradas no estabelecimento.

A decisão pela terceirização da execução de ações como fazer uma ligação telefônica e mostrar a posição geográfica num mapa foi tomada pensando na estrutura da plataforma do Android e em suas camadas de aplicações. Segundo dito na seção

⁴² <https://developer.android.com/reference/android/widget/ImageButton.html>

sobre o sistema de aplicativos em ANDROID (2016b), os aplicativos nativos ou desenvolvidos por terceiros que estão instalados no *smartphone* podem ter algumas de suas partes reaproveitados por outros aplicativos. Sendo assim, foram aproveitadas as soluções prontas e mais elaboradas e especializadas para realização de tais ações descritas acima.

5.2.4 A tela de configurações

A tela de configurações é utilizada para duas ações que influenciam no funcionamento central do aplicativo. É nela que o usuário deverá definir qual a distância máxima em quilômetros os seus resultados de busca por localização geográfica irão retornar. Nela também são definidos quais os municípios do Estado de Santa Catarina deverão aparecer nos resultados das buscas por parâmetros.

Todas as configurações do aplicativo podem ser alteradas acessando a tela principal do aplicativo onde encontra-se os estabelecimentos favoritos. Sua configuração é obrigatória para o funcionamento do aplicativo, tanto que durante a primeira execução do aplicativo no *smartphone* ela é a primeira tela que surge para o usuário após a tela de apresentação, como pode ser visto na Figura 28.

Figura 28 - Tela de configurações



Fonte: Elaborado pelo autor

Nesta primeira execução do aplicativo a configuração inicial é feita automaticamente para o usuário. São utilizadas as coordenadas geográficas do *smartphone* para descobrir qual o município atual ele se localiza e essa informação é definida nas configurações. A distância máxima para busca por estabelecimentos via GPS é definida por padrão como sendo 1 quilometro. O usuário pode alterar todas estas informações a qualquer momento, inclusive neste momento inicial.

Para adquirir as coordenadas geográficas da localização atual do usuário nesta primeira tela de configuração inicial, utilizou-se a classe *PrimeiraConfiguracaoActivity*. Esta classe herdou as características da classe *AppCompatActivity*⁴³, que pertence ao pacote da API de compatibilidade com telefones mais antigos, e implementa as interfaces: *ConnectionCallbacks*⁴⁴,

⁴³ <https://developer.android.com/reference/android/support/v7/app/AppCompatActivity.html>

*GoogleApiClient.OnConnectionFailedListener*⁴⁵, *LocationListener*⁴⁶, *ResultCallback<LocationSettingsResult>*⁴⁷. Todas essas interfaces auxiliam na instanciação do objeto *GoogleApiClient*⁴⁸ cuja principal função é a verificação se o *smartphone* possui o *Google Play Services* instalado, se ele está atualizado para uma versão que seja compatível com a versão do SDK que está sendo utilizado no desenvolvimento do aplicativo, se ele possui o serviço de localização no dispositivo, através de um GPS físico ou via internet, se este serviço está ativo e se o aplicativo possui autorização para utilizá-lo. Para cada uma dessas verificações o objeto executa uma ação, um método de *call-back* caso o resultado seja negativo para alguma delas. Caso todas as etapas sejam aprovadas o Android inicializa o processo de captura das coordenadas geográficas e através do método de *call-back* *onLocationChanged(Location location)*⁴⁹ ele entrega ao desenvolvedor um objeto do tipo *Location*⁵⁰ que possui atributos como latitude e longitude. Este método é executado repetidas vezes durante um espaço de tempo que podemos definir. Como são necessárias apenas as coordenadas atuais, foi configurado para que ele seja executado apenas uma vez. Com as coordenadas do usuário, uma chamada a um método privado da classe principal que executa operações em segundo plano é feita. A classe *CarregadorDeDados* que herda funcionalidades de *AsyncTask*⁵¹ realiza as seguintes ações para descobrir qual o município o usuário se encontra atualmente:

- Realiza uma chamada ao banco de dados retornando todos os estabelecimentos cadastrados;
- Calcula a distância em metros de cada um deles através do método *Location.distanceBetween(latitude,longitude,lat,lon,distancia)* que recebe como parâmetro dois pares de coordenadas, calcula a distância entre elas em métodos e armazena tal informação em uma atributo de cada objeto

<https://developers.google.com/android/reference/com/google/android/gms/common/api/GoogleApiClient.ConnectionCallbacks>

⁴⁵

<https://developers.google.com/android/reference/com/google/android/gms/common/api/GoogleApiClient.OnConnectionFailedListener>

⁴⁶ <https://developers.google.com/android/reference/com/google/android/gms/location/LocationListener>

⁴⁷

<https://developers.google.com/android/reference/com/google/android/gms/common/api/ResultCallback>

⁴⁸

<https://developers.google.com/android/reference/com/google/android/gms/common/api/GoogleApiClient>

⁴⁹

<https://developer.android.com/reference/android/location/LocationListener.html>

⁵⁰ <https://developer.android.com/reference/android/location/package-summary.html>

⁵¹ <https://developer.android.com/guide/components/processes-and-threads.html>

Estabelecimento;

- A lista de objetos do tipo estabelecimento é organizada em ordem crescente através do Classe *Collections*, que provê estruturas de dados do tipo *ArrayLists*, baseando-se no atributo distância.
- O primeiro elemento desta lista organizada é o que possui o município no qual o usuário está nesse momento.
- O nome do município é definido na tela de configuração inicial.

Com apenas um município já definido o usuário pode salvar os dados clicando no botão no canto direito superior da tela. Assim ele poderá retornar a tela inicial.

A tela de configuração utiliza componentes prontos contidos no *Preference Framework*⁵² disponibilizados pelo SDK do Android, na qual o componente da interface gráfica prontos para utilizar e montar telas padronizadas podem ser utilizadas para criar uma tela de configuração para o aplicativo. Ao utilizar esse framework percebe-se que ele é um pouco limitado em relação aos componentes pré-programados, porém é possível desenvolver componentes mais avançados como por exemplo uma *SeekBar*⁵³ que possa ser acoplada ao *framework* para que funcione de forma automatizada como os outros componentes. Toda a funcionalidade de salvamentos dos valores das configurações é transparente ao usuário do *framework*, por exemplo quando o usuário desliza a *seekbar* para definir a distância máxima que deseja visualizar estabelecimentos buscados via GPS, esse valor é salvo automaticamente pelo *framework* pois ele utiliza a mesma interface *SharedPreferences* que foi utilizada de forma separada na funcionalidade de adição de estabelecimentos de saúde ao arquivo XML de favoritos que é salvo na memória interna reservada ao aplicativo dentro da memória principal do *smartphone* como pôde ser visto na Figura 22. Já na Figura 29 é apresentado *layout* que gera a interface gráfica da tela de configuração do aplicativo.

Figura 29 - Declarações para geração do Layout da tela de configurações

⁵² <https://developer.android.com/guide/topics/ui/settings.html>

⁵³ <https://developer.android.com/reference/android/widget/SeekBar.html>

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
3      android:title="Configurações">
4
5      <PreferenceCategory android:title="Municípios">
6          <MultiSelectListPreference
7              android:defaultValue="@array/municipios_preferidos_default"
8              android:dialogTitle="selecione os municípios"
9              android:entries="@array/nomes_municipios_view"
10             android:entryValues="@array/codigos_municipios"
11             android:key="municipios_preferidos"
12             android:selectable="true"
13             android:summary="Não há municípios selecionados"
14             android:title="Municípios escolhidos:" />
15        </PreferenceCategory>
16
17        <PreferenceCategory android:title="Estabelecimentos">
18            <br.com.jucemar_dimon.cadesaude.SeekBarPreference
19                android:defaultValue="1"
20                android:key="@string/pref_distancia_estabelecimentos_key" />
21        </PreferenceCategory>
22
23        <br.com.jucemar_dimon.cadesaude.InfoPreference
24            android:key="info"
25            android:selectable="false"
26            android:enabled="false" />
27
28    </PreferenceScreen>

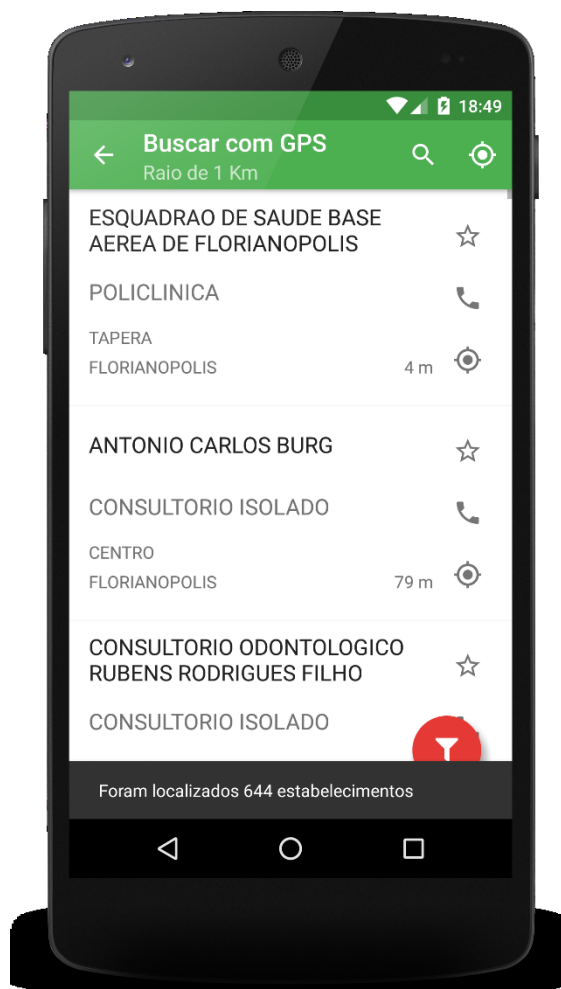
```

Fonte: Elaborado pelo autor

5.2.5 A tela de busca por GPS

A tela de busca por estabelecimentos através da posição geográfica do usuário é a funcionalidade principal do aplicativo. As buscas feitas nesta funcionalidade levam em consideração somente a distância máxima definida pelo usuário na tela de configurações do aplicativo para que o estabelecimento seja mostrado ou não nos resultados de busca. A Figura 30 faz a ilustração da interface gráfica da tela.

Figura 30 - Tela de busca por GPS



Fonte: Elaborado pelo autor

A tela mostra em seu cabeçalho qual o raio de distância predefinido nas configurações do aplicativo. Possui dois botões de ações na barra superior de menu: o filtro por nome e a nova busca por GPS para que o usuário possa atualizar sua pesquisa caso tenha mudado de posição geográfica. Assim como nas outras telas do aplicativo o componente visual utilizado para apresentar o resultado da busca pelos estabelecimentos é uma *RecyclerView*. Dentro dela são adicionados os estabelecimentos e seus dados básicos, como nome fantasia, bairro e cidade são apresentados juntamente com os botões de ações já explicado em seções anteriores. Nesta tela existe um botão de ação flutuante que se expande como um menu revelando outras diversas opções. Este componente visual é denominado de *FloatActonMenu*⁵⁴ e foi utilizado para auxiliar na criação de um filtro dos resultados encontrados. O *FloatActonMenu* é uma implementação personalizada que agrupa,

⁵⁴ <https://github.com/Clans/FloatingActionButton#floating-action-menu>

cria animações e interatividade entre os *FloatActonButton* que estão disponíveis como classes nativas do SDK do Android. O *FloatActonMenu* está disponível na biblioteca *FloatActonButton* desenvolvida por Dmytro Tarianyk⁵⁵ a qual foi utilizada neste projeto por permitir de uma forma muito transparente e prática a criação de menus flutuantes. As opções para filtrar os estabelecimentos são montadas de acordo com as características dos estabelecimentos retornados da busca. A Figura 31 demonstra todos os tipos de filtros possíveis.

Figura 31 - Filtros da tela de busca por GPS



Fonte: Elaborado pelo autor

Cada um dos tipos de filtros possui seus valores a serem filtrados baseando-se nas características dos estabelecimentos que resultaram na busca. Caso o ícone do tipo de filtro fique desabilitado subentende-se que só exista um valor de opção

⁵⁵ <https://github.com/Clans>

possível para ele, logo não existe sentido utilizá-lo.

O processo de busca por localização geográfica utiliza também a *GoogleApiClient*. Inicialmente é feita a verificação no dispositivo se existe a aplicação Google API Service instalada, isto é feito através do comando *isGooglePlayServicesAvailable*⁵⁶ que retorna um código que representa o sucesso ou fracasso, caso seja fracasso uma janela é apresentada ao usuário solicitando que ele instale tal serviço. A segunda etapa é construir um objeto com auxílio da classe *GoogleApiClient* e suas interfaces e métodos de *callback*:

- *onConnected*: faz a conexão do *GoogleApiClient* com o serviço de localização do dispositivo.
- *onRequestPermissionsResult*: Caso o usuário não tenha permissão para utilizar o serviço de localização no dispositivo, esse método abre uma janela solicitando que o usuário autorize o aplicativo a usar o serviço.
- *adquirirLocalizacao*: Esse método é chamado caso todas as etapas anteriores tenham sido aceitas, ele inscreve o aplicativo ao serviço de *Location* do Android para ser informado sobre as coordenadas a cada período que foi definido no objeto instanciado através da *GoogleApiClient*.
- *onLocationChanged*: quando este *callback* método é chamado ele já traz um objeto do tipo *Location* que contém a latitude e a longitude do atual *smartphone*.

Após adquiridas as coordenadas geográficas atuais do dispositivo, é utilizado uma classe que executa uma tarefa em segundo plano chamada através da classe *CarregadorDeDadosBGPSA* que herda as propriedades da classe *AsyncTask* permitindo executar a tarefa em uma *thread* separada da *UI thread*. Neste método é adquirida uma instância da classe *EstabelecimentosDAO* que utiliza os padrões de projeto DAO e *Singleton* para facilitar o acesso ao banco de dados da aplicação. Tendo esta instância única, o método *buscaEstabelecimentosPorGPS* que recebe com parâmetros as variáveis *localizacao* e *distanciaMaxima* é utilizado para retornar uma coleção do tipo *ArrayList* de objetos do tipo *Estabelecimento*. Esta coleção de objetos é gerada utilizando a seguinte sequência de passos:

⁵⁶

<https://developers.google.com/android/reference/com/google/android/gms/common/GoogleApiAvailability>

- Uma consulta ao banco de dados que retorna todos os estabelecimentos que possuem valores válidos nos campos Latitude e Longitude é realizada;
- Armazena-se estes dados em um *ArrayList* de objetos do tipo *Estabelecimento*, porém temporário;
- O método estático da classe *Location* da API do Android é utilizado para calcular a distância entre as coordenadas geográficas do estabelecimento e as coordenadas geográficas do dispositivo.
 - *Location.distanceBetween(localizacaoAtual.getLatitude(),localizacaoAtual.getLongitude(),latEstab,lonEstab,distancia);*
- O valor dessa distância é armazenado no atributo distância de cada objeto da classe estabelecimento contido na lista temporária.
- Este processo é feito para todos os estabelecimentos do banco e logo após eles são organizados em ordem de distância do menor para o maior.
- Na sequência é feita uma verificação de quais estabelecimentos possuem sua distância dentro da distância limite definida nas configurações do aplicativo e as que ficam dentro dos limites são armazenadas em uma terceira lista temporária que será retornada pelo método.

Com a lista de estabelecimentos configurada, a sua adição dentro da *RecyclerView* é realizada através de um adaptador. Todo esse processo é feito dentro do *call-back* método *onPostExecute* que é reescrito no método para que se possa criar uma comunicação com a *UI Thread* da interface gráfica e adicionar os estabelecimentos dentro da *RecyclerView*.

Quanto aos filtros possíveis de serem utilizados sobre os resultados, eles são aplicados filtros sobre filtros. Aplicando um filtro, o resultado deste na lista de estabelecimentos torna-se a base para a aplicação do próximo, e assim sucessivamente, até esgotarem-se as possibilidades de opções de filtros como pode ser visualizado na Figura 32.



Fonte: Elaborado pelo autor

O processo de filtragem dos estabelecimentos é feito dentro do método *doInBackground* da classe *CarregadorDeDadosBGPSA*. Os estados de todos os filtros são salvos em listas do tipo *ArrayList* e todas são salvas no contexto do aplicativo através do método *onSaveInstanceState*⁵⁷, para caso alguma outra tela do próprio aplicativo ou de outra aplicação do Android passe a ocupar o topo da tela do dispositivo. Todo o estado dos filtros aplicados são recuperados quando a tela de busca com GPS volta a ser visualizada através da instância do objeto da classe *Bundle* que pode ser acessada através do método de *call-back* *onSaveInstanceState* que recebe como parâmetro um objeto deste tipo.

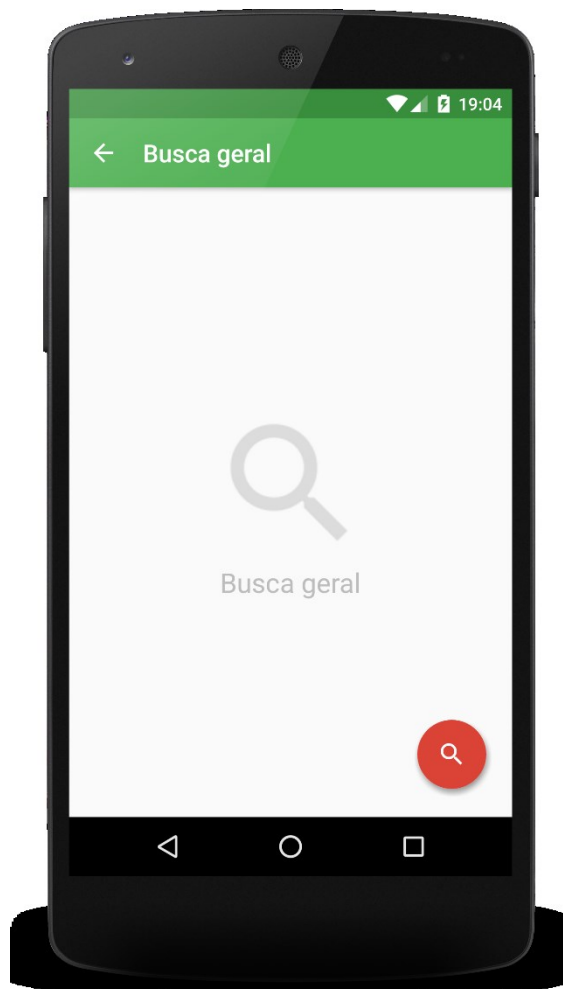
5.2.6 A tela de busca parametrizada

⁵⁷

[https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))

A tela de busca parametrizada é acessada através da tela principal do aplicativo tocando no botão de ação flutuante contendo o ícone de lupa. Esta funcionalidade do aplicativo leva em consideração no processo de busca por estabelecimentos os municípios predefinidos pelo usuário na tela de configuração do aplicativo. Tal decisão foi tomada pensando na performance das buscas no banco de dados pois utilizando-se do município como parâmetro de busca retornou-se uma quantidade menor de resultados nas consultas. Isto tornou a manipulação dos dados na memória mais prática. O objetivo de desenvolver esta funcionalidade foi pensado para permitir ao usuário uma busca de forma rápida por algum parâmetro específico que não possa ser buscado na função de busca por GPS. Um contexto que exemplifica a necessidade de tal funcionalidade seria: um usuário que consultou em uma data passada com um profissional médico e alguns meses depois essa paciente descobre que este profissional não atende mais na clínica na qual ocorreu o atendimento anterior, porém este paciente prefere por insistência consultar com este mesmo profissional por ter sentido satisfação em seu trabalho. Utilizando a função de busca pelo nome do profissional nesta funcionalidade do aplicativo é possível descobrir todos os estabelecimentos que este profissional está trabalhando no momento. Este é apenas um exemplo de contexto de como essa funcionalidade do aplicativo pode ser útil, uma outra muito interessante seria a busca de uma clínica por parte do nome fantasia ou razão social do estabelecimento. A Figura 33 ilustra a tela de busca por parâmetros.

Figura 33 - Tela de busca por parâmetros



Fonte: Elaborada pelo autor

Esta é a tela qual o usuário fará sua busca por estabelecimentos utilizando parâmetros que serão selecionáveis através da janela de diálogo que será apresentada no momento em que o botão de ação flutuante com ícone de lupa for tocado. A partir de então uma série de configurações de busca pode ser feita. Os tipos de buscas possíveis das quais uma delas deve ser selecionada inicialmente na janela de diálogo são:

- Tipos de convênios: São as possíveis formas de pagamentos pelo atendimento prestado pelo estabelecimento, esses valores são predefinidos e podem ser:
 - Particular;
 - Plano / Seguro próprio;
 - Plano / Seguro terceiro;
 - Plano de saúde privado;
 - Plano de saúde público;

- SUS.
- Formas de atendimento: São as formas com as quais os estabelecimentos trabalham para atender aos clientes segundo sua estrutura física e estrutura e profissional:
 - Ambulatorial (Atendimentos com pouca urgência);
 - Internação (Atendimentos com pouca ou muita urgência que necessitem de acompanhamento monitorado);
 - SADT (Serviço de exames para diagnósticos: Exames laboratoriais, Raios-X, eletrocardiogramas, entre outros);
 - Urgência (Atendimentos que envolvem risco de vida).
- Nome do profissional: Modalidade de busca pelo nome do profissional. O termo buscado deverá ser digitado pelo usuário;
- Especialidades: Neste tipo de busca será retornada à janela de diálogo uma lista completa com todos os tipos de especialidades médicas cadastradas nos estabelecimentos de saúde localizados nos municípios definidos pelo usuário para realizar a busca;
- Nome do estabelecimento: Modalidade de busca na qual o usuário deverá digitar um termo a ser buscado;
- Serviços oferecidos: Modalidade de busca onde o usuário deverá selecionar um tipo de serviço especializado oferecido pelos estabelecimentos dos municípios listados para serem inclusos nos resultados de busca na tela de configurações do aplicativo. Os tipos de serviços possíveis podem ser vistos no Anexo B.

Após realizada a configuração da busca o usuário iniciará a busca e os resultados serão listados na tela. A Figura 34 demonstra um exemplo.



Fonte: Elaborada pelo autor

Esta tela é processada pela classe *BuscaGeralActivity* que herda da classe nativa do Android *AppCompatActivity* e implementa uma interface personalizada que foi desenvolvida para criar um canal de comunicação entre esta classe e a janela de diálogo que servirá para configurar os parâmetros de busca. A interface criada foi nomeada de *InterfaceBuscaGeralPorString* e a mesma possui o método *informarTermoParaConsulta* que recebe como parâmetro um uma constante inteira chamada de *tipoBusca* que conterà o tipo de busca a ser feita e uma variável do tipo *String* chamada de *termo* que armazenará qual o valor a ser buscado no banco de dados. Esse método *call-back* é chamada pelo Android no momento que a janela de configurações das pesquisas fecha.

As técnicas utilizadas para poder produzir ao usuário as configurações de todas essas funcionalidades são baseadas na classe *BuscaGeralDialog*. Esta classe herda as características da classe nativa do Android, a *DialogFragment* e implementa uma interface que permite a interação e vinculação entre os dois campos de lista

suspensa do tipo *Spinner*⁵⁸ nos quais os tipos possíveis de buscas são apresentados, a *AdapterView.OnItemSelectedListener*⁵⁹. Esta interface permitiu que os dois elementos de interface gráfica do tipo *Spinner*: *comboBoxTipoBusca* e *comboBoxBusca* pudessem interagir de forma que dependendo do valor que o usuário selecionou em um influenciasse nas opções carregadas no outro. Todas estas opções carregadas nestes elementos, exceto as opções de tipos de busca que estão armazenadas em um simples *Array* de *Strings*, são advindas do banco de dados. Dessa forma para que o carregamento desses dados não interferissem nos processos que a *UI Thread* está executando, criou-se uma classe privada nomeada e *CarregadorDeDados*. Esta classe herda as propriedades da classe nativa do Android, a *AsyncTask*, que executa os procedimentos definidos em seus métodos em uma *thread* em segundo plano e de forma assíncrona. A interface *AdapterView.OnItemSelectedListener* foi vinculada aos elementos objetos *comboBoxTipoBusca* e *comboBoxBusca*. Dessa forma quando o usuário interage e define um tipo de busca em *comboBoxTipoBusca* o Android se utiliza da interface que foi implementada para chamar o método *call-back onItemSelected*. Neste método a seleção feita pelo usuário no *comboBoxTipoBusca* é capturada e então uma instância da classe *CarregadorDeDados* é criada para executar os métodos que farão as consultas no banco de dados. A Figura 35 ilustra esta execução.

Figura 35 - Trecho de código que representa as consultas ao banco

⁵⁸ <https://developer.android.com/guide/topics/ui/controls/spinner.html>

⁵⁹ <https://developer.android.com/reference/android/widget/AdapterView.OnItemSelectedListener.html>

```

226      @Override
227      protected Void doInBackground(Void... params) {
228          listaDeMunicipios = PreferenceManager.getDefaultSharedPreferences(getActivity())
229              .getStringSet("municipios_preferidos", new HashSet<String>());
230          switch (tipoDeBusca) {
231              case SPINNER_CONVENIOS_ACETOS:
232                  AtendimentoDAO atendimentoDAOConv = AtendimentoDAO
233                      .getInstancia(contexto.getApplicationContext());
234                  dadosSpinner02 = atendimentoDAOConv.listarTiposConvenios(listaDeMunicipios);
235                  break;
236              case SPINNER_ESPECIALIDADES:
237                  ProfissionaisDAO profissionaisDAO = ProfissionaisDAO
238                      .getInstancia(contexto.getApplicationContext());
239                  dadosSpinner02 = profissionaisDAO.listarTiposProfissionais(listaDeMunicipios);
240                  break;
241              case SPINNER_FORMAS_DE_ATENDIMENTO:
242                  AtendimentoDAO atendimentoDAOAtend = AtendimentoDAO
243                      .getInstancia(contexto.getApplicationContext());
244                  dadosSpinner02 = atendimentoDAOAtend.listarTiposAtendimento(listaDeMunicipios);
245                  break;
246              case SPINNER_SERVICOS_OFERECIDOS:
247                  ServicoClassificacaoDAO servicoClassificacaoDAO = ServicoClassificacaoDAO
248                      .getInstancia(contexto.getApplicationContext());
249                  dadosSpinner02 = servicoClassificacaoDAO.listarTiposServicos(listaDeMunicipios);
250                  break;
251          }
252          return null;
253      }

```

Fonte: Elaborado pelo autor

Na figura acima percebe-se que a variável *listaDeMunicipios* armazena os municípios recuperados que foram definidos pelo usuário nas configurações do aplicativo. Na sequência verifica-se com uma estrutura de seleção o tipo de busca definido pelo usuário e dependendo desse valor é adquirida uma instância de um tipo específico de classe que manipula um tipo específico de dados no banco de dados do aplicativo. Após o resultado do método que traz as opções do tipo específico de consulta eles são carregados na interface gráfica através do método *onPostExecute* que é o responsável por solicitar a *UI Thread* que execute os comandos contidos nele para que o *comboBoxBusca* possa ser preenchido com as opções carregadas do banco de dados.

Definidos os parâmetros de busca, o usuário toca no botão confirmar então a janela de diálogo fecha. Nesse momento, o método *informarTermoParaConsulta* da interface *InterfaceBuscaGeralPorString* é chamado e o foco passa a ser a tela principal de busca pois a janela de diálogo será fechada.

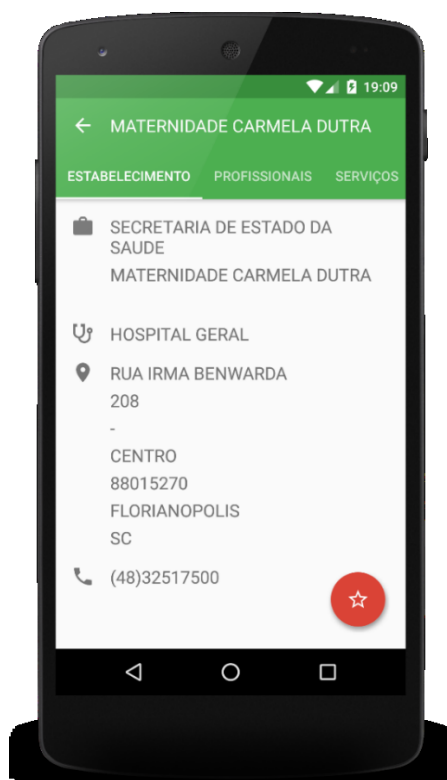
A partir desse momento já existe a informação sobre o tipo de busca e o valor a ser buscado nos estabelecimentos. Neste momento que o método *call-back* da interface é chamado, usa-se uma classe para executar em segundo plano a busca por esses estabelecimentos que possuem esse parâmetro em alguma de suas características. Após a busca feita, um *ArrayList* de objetos do tipo *Estabelecimento* é

retornado e repassado ao adaptador da *RecyclerView* para que esses estabelecimentos sejam mostrados na tela. Estes estabelecimentos mostrados dentro da *RecyclerView* possuem todas as funcionalidades das listas das outras telas como: funcionalidades de adição do estabelecimento a lista de favoritos, funcionalidade de visualização da posição geográfica em um aplicativo de mapa terceiro e a funcionalidade realizar ligações para o estabelecimento.

5.2.7 A tela de visualização do estabelecimento

A tela de visualização dos dados dos estabelecimentos é acionada a partir do toque em algum estabelecimento contido na *RecyclerView*. Quando um item dessa estrutura é tocado uma nova tela é aberta contendo todos os dados contidos no banco de dados sobre o estabelecimento em questão: Nome fantasia; Razão social; Endereço; Telefone; Profissionais que atuam no estabelecimento; Serviços; Categorias de atendimento e Convênios. Todas estas informações são apresentadas na tela numa estrutura de guias deslizáveis. A Figura 36 ilustra esta tela.

Figura 36 - Tela de detalhes do estabelecimento



Fonte: Elaborado pelo autor

A guia Estabelecimentos que mostra todos os dados principais referentes ao

estabelecimento ainda possui algumas funcionalidades para facilitar o uso pelo usuário. Ele também pode ter acesso à função de ligação direta, clicando no ícone do telefone, visualização da localização do estabelecimento em um mapa ao clicar no ícone de marcação de mapa e também pode adicionar o estabelecimento a sua lista de favoritos clicando no botão de ação flutuante com ícone de estrela.

As técnicas que permitem a visualização desta tela aos usuários foram desenvolvidas utilizando com base a classe *EstabelecimentoActivity*. Esta classe personalizada herda as características da classe nativa do Android chamada *AppCompatActivity*. A estrutura de guias foi desenvolvida baseando-se na criação de uma classe personalizada nomeada de *VisualizadorDePaginasAdapter* que herda as características da classe *FragmentStatePagerAdapter*⁶⁰. Essa classe permite de forma simples a inserção de *Fragments*⁶¹, que são na verdade uma forma simplificada de telas as quais podem ser agrupadas e apresentadas na tela do dispositivo de várias formas diferentes. Para permitir essa estrutura de guias foram criadas várias classes que herdaram as características desses *Fragments*:

- *AtendimentoFragment*;
- *EstabelecimentoFragment*;
- *ProfissionaisFragment*;
- *ServicosFragment*.

Cada um desses fragmentos é adicionado ao *VisualizadorDePaginasAdapter* e através da guia nomeada com o título das informações contidas nela, o usuário navega através delas simplesmente tocando no título da guia ou arrastando as telas do aplicativo, pois a classe *FragmentStatePagerAdapter* providencia todos mecanismos necessários para tal. A Figura 37 mostra o arquivo de layout em formato XML que auxilia na criação a tela em questão.

Figura 37 - Layout da visualização em abas da interface gráfica

⁶⁰ <https://developer.android.com/reference/android/support/v4/app/FragmentStatePagerAdapter.html>

⁶¹ <https://developer.android.com/reference/android/app/Fragment.html>

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:fab="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:fitsSystemWindows="true"
9      tools:context="br.com.jucemar_dimon.cadesaude.EstabelecimentoActivity">
10
11      <android.support.design.widget.AppBarLayout
12          android:layout_width="match_parent"
13          android:layout_height="wrap_content"
14          android:theme="@style/MeuTemaMaterial.AppBarOverlay">
15
16          <android.support.v7.widget.Toolbar...>
17
18          <android.support.design.widget.TabLayout
19              android:id="@+id/tabs"
20              android:layout_width="match_parent"
21              android:layout_height="wrap_content"
22              app:tabGravity="fill"
23              app:tabIndicatorColor="@color/icons"
24              app:tabMode="scrollable"
25              app:tabSelectedTextColor="@color/icons"
26              app:tabTextColor="@color/primary_light" />
27      </android.support.design.widget.AppBarLayout>
28      <android.support.v4.view.ViewPager
29          android:id="@+id/viewpager"
30          android:layout_width="match_parent"
31          android:layout_height="match_parent"
32          android:background="@color/background_material_light"
33          android:visibility="gone"
34          app:layout_behavior="@string/appbar_scrolling_view_behavior" />
35  </android.support.design.widget.CoordinatorLayout>

```

Fonte: Elaborado pelo autor

O elemento *TabLayout*⁶² trata de gerar as guias nomeadas usadas pelo usuário para a abertura de cada uma das telas e o elemento *ViewPager*⁶³ é o responsável por armazenar e exibir cada uma das telas de *Fragments* a medida que forem chamadas.

Para o carregamento dos dados dos estabelecimentos em cada uma das telas é utilizado um processo em segundo plano disponibilizado através da classe privada *CarregadorDeDados* que herda as características da classe *AsyncTask* cujos comandos armazenados no método *doInBackground* são executados em uma *thread* separada da *UI Thread*. Para cada classe do tipo *Fragment* uma classe do banco de dados é utilizada:

- *AtendimentoFragment*: Utiliza a classe *AtendimentoDAO* para extrair do banco de dados um *ArrayList* de objetos do tipo *Atendimento* que será passado como argumento para que futuramente seja recuperado e apresentado na lista de atendimento quando o fragmento é exibido;
- *ProfissionaisFragment*: Utiliza a classe *ProfissionaisDAO* para extrair do

⁶² <https://developer.android.com/reference/android/support/design/widget/TabLayout.html>

⁶³ <https://developer.android.com/reference/android/support/v4/view/ViewPager.html>

banco de dados um *ArrayList* de objetos do tipo *Profissional* que será passado como argumento para que futuramente seja recuperado e apresentado na lista de profissionais quando o fragmento é exibido;

- *ServicosFragment*: Utiliza a classe *ServcosDAO* para extrair do banco de dados um *ArrayList* de objetos do tipo *ServicoClassificacao* que será passado como argumento para que após seja recuperado e apresentado na lista de atendimento quando o fragmento é exibido;

O objeto da classe *Estabelecimento* que contém as informações básicas é extraído do objeto *Bundle*⁶⁴ da *intent* da *EstabelecimentoActivity* que por sua vez recebe tal objeto da lista de estabelecimentos armazenados pela *RecyclerView*. Tendo o objeto da classe *Estabelecimento* e a lista contendo os convênios e as formas de atendimento, a lista contendo os profissionais vinculados e a lista contendo os serviços especializados, todos os campos da tela que descreve o estabelecimento podem ser completados seus gráficos e textuais.

5.2.8 A publicação na loja de aplicativos

Para facilitar ao usuário o processo de aquisição de novos aplicativos, a plataforma Android possui um sistema de filtros que é transparente aos usuários. Caso o desenvolvedor do aplicativo siga corretamente as instruções que são divulgadas na documentação oficial do Android acerca das configurações desses filtros em seu arquivo APK que é armazenado na loja de aplicativos do Google, esse sistema de filtros atingirá seu objetivo e somente permitirá aos usuários da *Google Play Store* instalar os aplicativos que são compatíveis com o seu aparelho. Caso o desenvolvedor não dispenda atenção suficiente sobre esta questão, pode haver um aumento significativo dos cenários em que o usuário instala um aplicativo através da *Google Play Store* e ele não funciona corretamente.

Esses sistemas de filtros são basicamente as declarações de elementos XML contendo as configurações desejadas dentro do arquivo *AndroidManifest.XML*. Os elementos utilizados para criação de filtros são baseados nas propriedades possíveis das declarações *<uses-feature>*⁶⁵. Com estas declarações podem ser definidos quais

⁶⁴ <https://developer.android.com/reference/android/os/Bundle.html>

⁶⁵ <https://developer.android.com/guide/topics/manifest/uses-feature-element.html>

os componentes de hardware e software um dispositivo deve ter para ser compatível com o aplicativo que se está declarando. Sendo assim, somente se este dispositivo tiver tais características ele poderá receber tal aplicativo na *Google Play Store*. Para o aplicativo que está sendo desenvolvido formam declaradas as características obrigatórias definidas na Figura 38.

Figura 38 - definição das Uses Feature no arquivo de manifesto

```
<uses-feature
    android:name="android.hardware.screen.portrait"
    android:required="true" />
<uses-feature
    android:name="android.hardware.telephony"
    android:required="true" />
<uses-feature
    android:name="android.hardware.location.gps"
    android:required="true" />
```

Fonte: Elaborado pelo autor

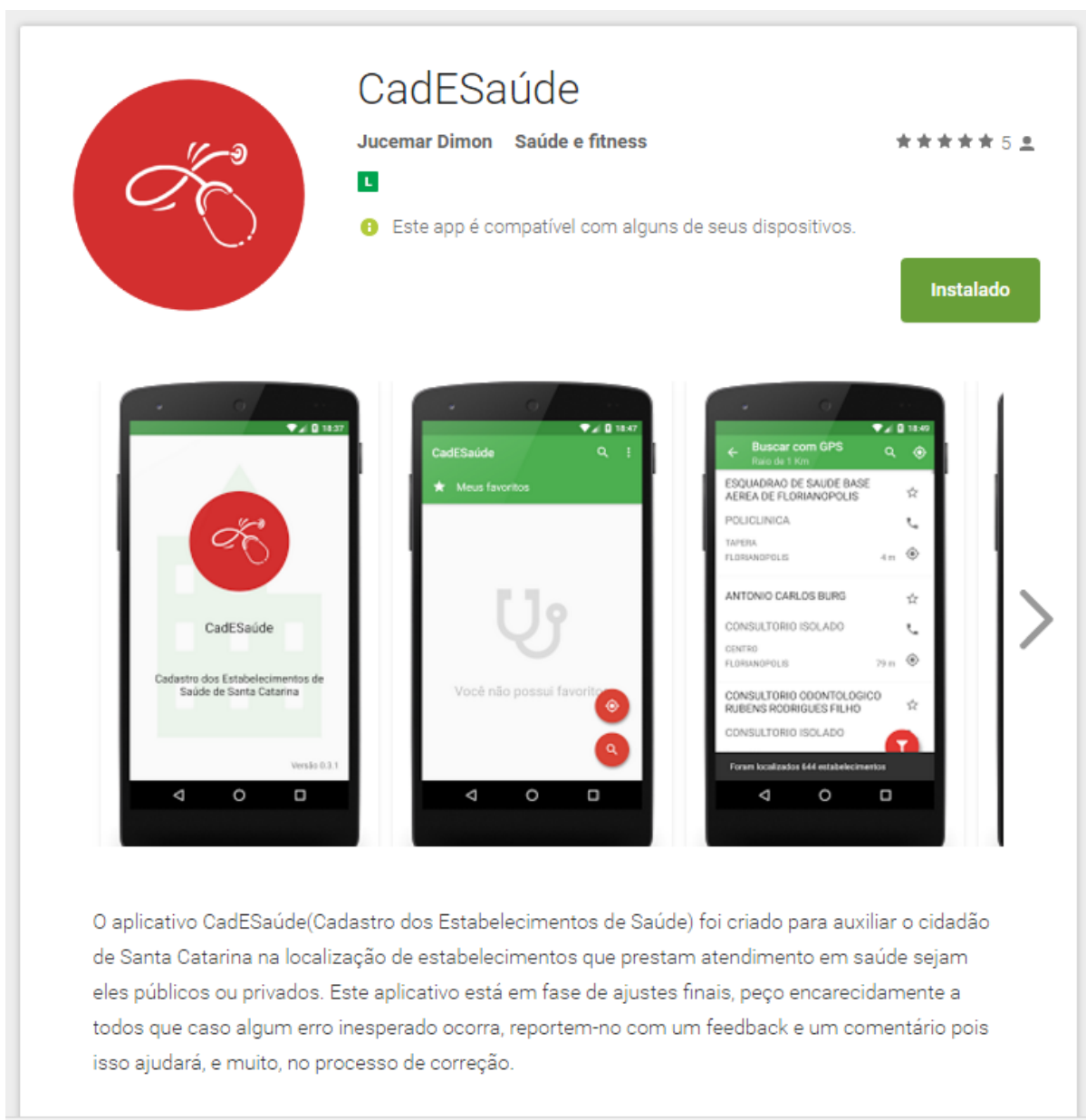
Segundo declarado na figura acima, para que o aplicativo seja compatível com o dispositivo ele somente necessita possuir um GPS, possuir a capacidade de realizar ligações telefônicas e poder utilizar a tela do dispositivo em modo vertical. Porém outras preocupações foram tomadas para garantir que além de compatível segundo os requisitos declarados, o aplicativo também funcionasse de forma correta.

Uma grande questão que poderia auxiliar no bom funcionamento do aplicativo no dispositivo no qual ele será instalado são os testes. A IDE *Android Stúdio* Fornece diversas possibilidades de testes de aplicativos, uma das mais interessantes foi utilizada durante o desenvolvimento do projeto: *Android Virtual Device*. Esta funcionalidade permite criar dispositivos virtuais simulando grande parte das funcionalidades de dispositivos reais. Dessa forma, pode-se testar o comportamento do aplicativo nas mais diversas versões do Android nas mais diversas características de hardware de dispositivos como tamanho e resolução de tela e poder de processamento. Para os testes no aplicativo foram utilizados emuladores com as seguintes versões do Android: 4.0; 4.1.1; 4.3; 4.4.4; 5.0; 5.1 e 6.0.

Após a realização de todas as etapas de desenvolvimento, preparação do projeto, geração e assinatura digital do arquivo APK, a sua publicação na loja de

aplicativos do Google através do portal *Google Play Developer Console* foi realizada e o aplicativo foi disponibilizado ao público. A primeira publicação do aplicativo foi feita no dia 08 de maio de 2016. Em sua versão inicial 0.01. A partir de então a cada nova alteração ou correções de bugs uma nova versão do aplicativo era disponibilizada. Na Figura 39 pode ser vista a página de apresentação do aplicativo.

Figura 39 - Página do aplicativo na Google Play Store



Fonte: Elaborado pelo autor

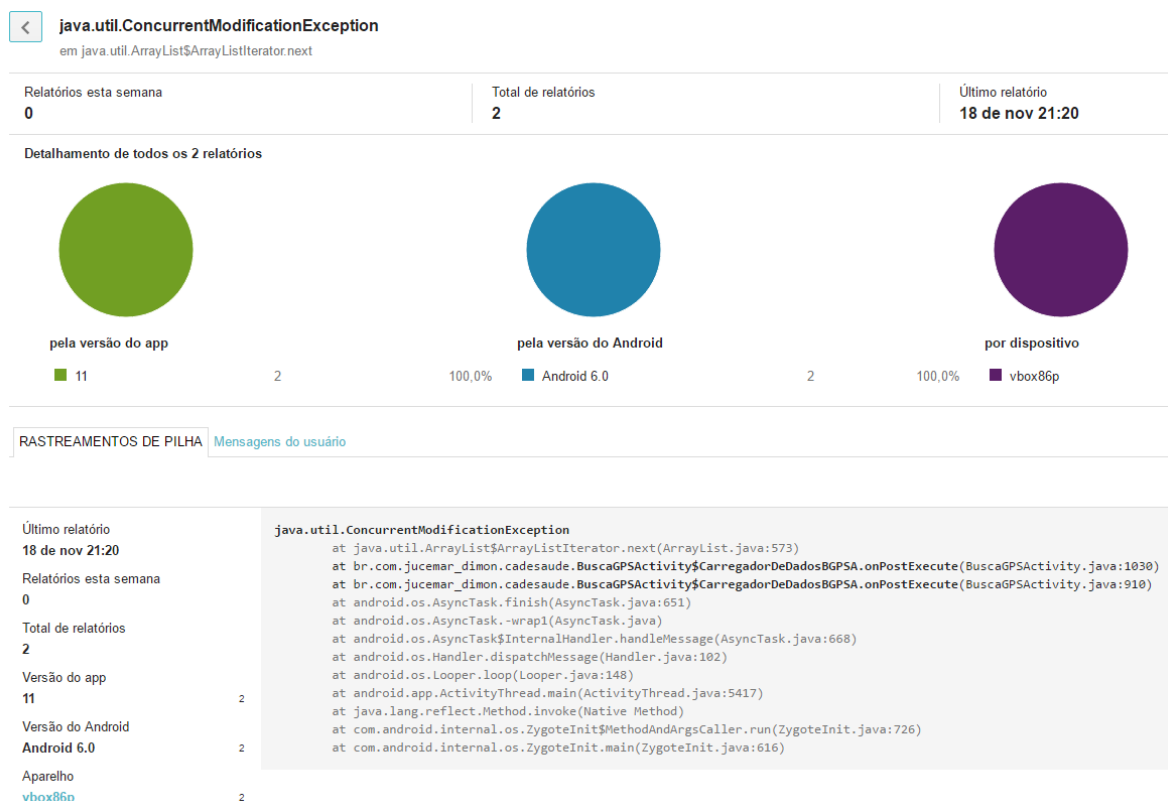
5.2.9 Avaliação do aplicativo

A avaliação do aplicativo pelos usuários foi feita de forma informal. Não foi feita uma divulgação em massa em nenhum veículo de comunicação ou qualquer outra do gênero sobre sua criação. Não se utilizou de nenhuma metodologia de avaliação da usabilidade da interface gráfica para testar o aplicativo. Foi feito apenas um simples teste na qual um pequeno grupo de pessoas foi convidado a realizar os simples atos de: entrar na página de divulgação do aplicativo, baixar, instalar, usar e reportar de suas impressões sobre o mesmo. As pessoas convidadas foram selecionadas segundo alguns requisitos:

- Selecionou-se pessoas com formações em áreas da saúde: Médicos, enfermeiras ou técnicos de enfermagem. Dessa forma caso algum termo técnico voltado a área da saúde fosse apresentado erroneamente, um respaldo seria dado;
- Pessoas com conhecimentos técnicos na área de desenvolvimento de aplicativos para a plataforma Android. Assim, caso alguma implementação de alguma ação dentro do aplicativo fugisse totalmente das práticas atuais utilizadas no desenvolvimento da plataforma um respaldo sobre sua correção o melhoramento seria oferecido;
- Pessoas com muita afinidade com a plataforma Android, que são usuários frequentes de smartphones e de seus aplicativos. Com isto, eles próprios poderiam realizar algum comparativo com outros aplicativos que usam frequentemente e reportar se o aplicativo CadESaúde foge aos padrões dos demais aplicativos.

Os convidados instalaram o aplicativo e retornaram suas impressões de forma informal. Alguns de forma verbal, outros através da avaliação do aplicativo em sua página de divulgação através de comentários ou classificações positivas. Foi possível também receber um retorno dos usuários contra falhas no aplicativo durante sua utilização pois a plataforma Android fornece ao usuário a opção de reportar ao responsável pelo aplicativo um erro ocorrido durante sua execução. O responsável pelo aplicativo pode visualizar o relatório de erros através do console do desenvolvedor da loja de aplicativo do Google. A Figura 40 ilustra um reporte de um caso de erro.

Figura 40 - Feedback com relatório de erro



Fonte: Elaborado pelo autor

Ao longo do desenvolvimento do aplicativo, foram publicadas 10 versões novas contendo as correções de vários itens: erros reportados pelos convidados; novas ideias sugeridas; correções de erros encontrados pelo autor durante o desenvolvimento; atualizações dos dados do banco de dados; entre outros.

Da forma como foi projetado, o aplicativo exige uma nova versão a cada mês para que se possa atualizar os dados contidos no banco de dados do aplicativo para que fiquem de acordo com o que é divulgado no site do CNES⁶⁶. Quanto à estratégia de lançamento das novas versões, pensou-se em ser feita logo após o quinto dia útil de cada mês. Essa decisão foi tomada baseado no cronograma do CNES⁶⁷, pois, geralmente, esse período é a data limite para que os municípios enviem ao banco de dados nacional as alterações feitas nos cadastros. Sendo assim uma única atualização mensal seria necessária para tanto corrigir bugs descobertos quanto a atualização do banco de dados. A atualização do banco de dados consiste em

⁶⁶

http://cnes2.datasus.gov.br/Lista_Tot_Es_Municipio.asp?Estado=42&NomeEstado=SANTA%20CATARINA

⁶⁷ <http://cnes.datasus.gov.br/pages/acesso-rapido/cronograma.jsp>

executar a ferramenta de captura de dados na data estipulada, gerar uma nova base de dados, substituir a anterior pela atual no projeto da IDE *Android Studio*, preparar o projeto para exportação com a nova versão e fazer o upload do arquivo APK no console de desenvolvedor da Google para que o mesmo promova a notificação sobre uma nova atualização aos usuários que possuem o aplicativo instalado.

5.2.10 Análise geral do uso da memória

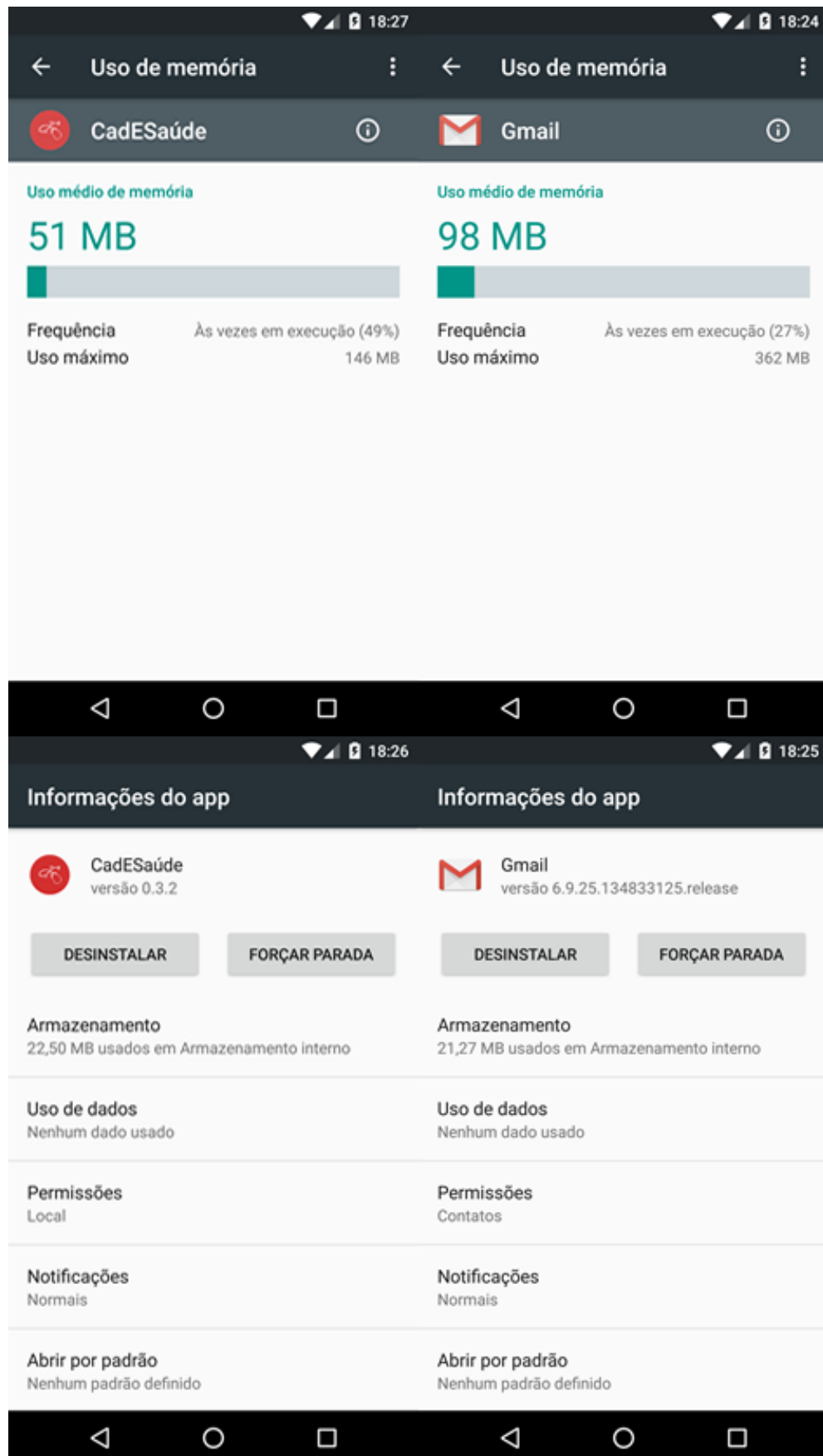
Uma das últimas versões da plataforma Android lançada, o Android 6.0 conhecida popularmente como *Android Marshmallow*, trouxe algumas novas ferramentas, e o melhoramento de diversas outras, para gerenciamento e controle do dispositivo. Uma destas ferramentas melhoradas foi a responsável pelo monitoramento da memória RAM dos aplicativos em execução, sua nova versão armazena o registro do uso de memória RAM de cada aplicativo possibilitando assim o monitoramento do gasto de memórias das aplicações em separado e também como um todo podendo assim saber o quanto de memória livre o dispositivo ainda possui (JUNIOR, 2016).

O objetivo desta seção é realizar um teste básico de memória do aplicativo desenvolvido. O aplicativo é totalmente off-line e possui um banco de dados interno frequentemente utilizado durante sua execução. Dessa forma são utilizados muitos dos recursos da memória RAM do dispositivo para auxiliar nos processos de consulta ao banco. Com esse pequeno teste pode-se verificar a situação dos picos de consumo da memória RAM do aplicativo no dispositivo. Assim, caso os picos de consumo de memória cheguem a valores que façam o aplicativo parar de responder, outras estratégias em relação a forma como os resultados das consultas ao banco que são armazenadas em memória deverão ser utilizadas.

Para que fosse possível verificar o comportamento do aplicativo em relação ao seu consumo de memória foi utilizado um emulador com as características do *smartphone LG NEXUS 5* que podem ser conferidas no site da própria fabricante em LG (2016). A memória RAM que este modelo de telefone possui é de 2GB e para este teste de memória foi utilizado o gerenciador de memória do *Android Marshmallow*. O aplicativo *CadESaúde*, desenvolvido neste trabalho, foi aberto no emulador e ficou ativo durante 3 horas e em tempos aleatórios o seu uso era feito. Para que se pudesse ter algum parâmetro para comparação, utilizou-se um aplicativo popular para que

também participasse do teste. O aplicativo utilizado foi o *Gmail*. Este aplicativo foi escolhido pois ele é nativo da plataforma Android. Ele foi desenvolvido pela própria Google que também é responsável pela plataforma. Pode-se interpretar que foi bem desenvolvido e que se preocupa com questões como o consumo de memória RAM de recursos de processamento. Logo, pareceu ser a melhor opção de aplicativo a se comparar. Sendo assim, se os resultados da comparação mostrarem que o CadESaúde possui valores maiores do que o *Gmail* em relação ao consumo de memória RAM, alguma ação corretiva deveria ser tomada. Ambos os aplicativos foram iniciados e ficaram ativos durante todo o tempo do teste. A Figura 41 abaixo exibe o resultado.

Figura 41 - Comparativo do consumo de memória RAM



Fonte: Elaborado pelo autor

Visualizando os valores nas telas capturadas no emulador na parte superior da Figura percebe-se que o *CadESaúde* possui um consumo médio de memória menor que o *Gmail*. Ambos aplicativos testados não são tipos de aplicativo nos quais são utilizados durante horas seguidas. O *Gmail*, por exemplo, é um aplicativo para enviar e receber e-mails, possivelmente um usuário recebe uma notificação de nova mensagem, abre o aplicativo lê a mensagem e na sequência fecha o mesmo. O *CadESaúde* também é um tipo de aplicativo no qual o usuário vai abri-lo, procurar o estabelecimento que necessita e na sequência, possivelmente, irá fechar o aplicativo. Durante o seu uso podem acontecer picos de utilização da memória RAM como foi observado na figura, o aplicativo *CadESaúde* teve um uso máximo de memória em algum momento do tempo em que foi realizado o teste, porém sua média de utilização é menor se comparada com o aplicativo do Gmail.

Observando os dados das telas capturadas do emulador percebe-se na parte inferior da figura os valores dos armazenamentos internos de cada um dos aplicativos. *CadESaúde*, apesar de possuir um banco de dados interno contendo aproximadamente 15.000 estabelecimentos de saúde armazenado em um arquivo, ocupa somente 22,50MB da memória interna do aplicativo.

6 CONCLUSÕES

O projeto desenvolvido finda com a produção da aplicação de captura de dados e com a publicação do aplicativo de busca por estabelecimentos na loja de aplicativos da Google.

O desenvolvimento de uma ferramenta de captura de dados para adquirir as informações sobre os estabelecimentos de saúde que foi apresentado na seção 1 do capítulo 4. As experiências vividas nesta etapa do projeto relembram a questão da importância da qualidade dos dados. Mesmo com todo o embasamento legislativo e suporte aos sistemas de saúde no DATASUS, ainda assim, foram encontrados casos de dados mal formatados, textos e campos de textos que não seguem um padrão de escrita. Um exemplo desse fato foi a falta do número de contato telefônico de alguns estabelecimentos. Uma informação como esta é de vital importância não só no contexto deste trabalho. Também tiveram casos em que o campo de telefone continha um número, porém ele, ou estava mal formatado, ou faltando um dígito. Outra questão que penalizou o resultado do trabalho foi quantidade de erros nos nomes dos logradouros de alguns estabelecimentos. Tal fato produziu discrepâncias nos resultados de busca das coordenadas geográficas de alguns estabelecimentos. Isto causou divergências quanto a sua localização exata durante sua visualização em um mapa. Contudo, pode-se considerar atingido o objetivo específico número 2, pois mesmo produzindo informações um pouco imprecisas dos estabelecimentos, a ferramenta cumpre o papel que se propôs.

Os demais objetivos específicos podem ser definidos como atingidos. O objetivo específico 1, que tratou da criação de uma especificação de requisitos para o aplicativo, pode ser visualizado no Apêndice I do documento. O objetivo específico 3, que tratou do desenvolvimento de um aplicativo, pode ser visto em detalhes na seção 2 do capítulo 4. E, por fim, o objetivo específico 4, que trata da publicação do aplicativo na *Google Play Store*, pode ser visto na subseção 2.8 do capítulo 4.

Pode-se dizer que o objetivo geral do projeto também foi atingido, pois produziu-se algo que fica muito próximo do projetado. O atingimento dos objetivos específicos somados, foi capaz de produzir uma solução para auxiliar os cidadãos do Estado de Santa Catarina na busca por estabelecimentos de saúde. O fato de o aplicativo não necessitar de uma conexão com a internet, retirou dele um fator

limitador em aplicativos que a exigem, pois existem áreas interioranas onde a rede da operadora de telefonia não oferece uma cobertura adequada. Dessa forma o aplicativo poderia se tornar inoperante. Contudo, questões como o tempo de execução do processo da captura dos dados da ferramenta de captura e a falta de um padrão no formato dos dados dos estabelecimentos deixaram a desejar.

Os temas e tecnologias aqui abordados acrescentaram grande conhecimento para seu autor, pois uma vasta pesquisa da plataforma Android foi necessária para o desenvolvimento do aplicativo. O foco principal das técnicas necessárias para o seu desenvolvimento foram a manipulação da API de localização para obter acesso aos dados do GPS e da API de banco de dados para a realização de consultas. Um conhecimento aprofundado em HTML foi adquirido neste trabalho. Tal aquisição foi necessária pois a biblioteca *JSOUP* que foi utilizada na construção da ferramenta de captura de dados, utiliza a manipulação direta dos elementos dessa linguagem. Este conhecimento fez a diferença no momento da criação dos métodos que navegam nas páginas do portal do CNES pois estas não seguem o padrão W3C.

6.1 TRABALHOS FUTUROS

Esta seção fornece ao leitor algumas questões que podem ajudá-lo a identificar possibilidades de continuar os trabalhos desenvolvidos aqui, melhorando de forma geral o sistema em todos os seus aspectos.

6.1.1 Melhorias na ferramenta de captura de dados

Sugere-se a melhoria da ferramenta de captura de dados pois em seu estado atual ela utiliza-se de apenas uma *Thread* para realizar todos os processos de gerenciamento do banco de dados, requisições ao servidor do DATASUS, consumo de *webservices* para aquisição de *geocooding* de endereços dos estabelecimentos e leitura de arquivos de texto. Dessa forma, a realização de todas essas tarefas automatizadas demanda muito tempo.

6.1.2 Adaptação do aplicativo para o funcionar parcialmente online

Sugere-se a adequação do sistema para que possibilite a utilização do

aplicativo em todo o território nacional. Para tal deverá ser pesquisada uma técnica que possibilite tal alteração de forma que a performance não seja afetada. As alterações necessárias para o aplicativo Android trabalhar com Estados do Brasil diferentes seria a troca do arquivo de banco de dados. Sendo assim não há como criar a possibilidade de uma opção de “configurar” o aplicativo para outros estados pois no arquivo APK de instalação não caberiam todos os estados. A estrutura do sistema hoje só permite que haja um arquivo de banco de dados no aplicativo contendo dados de um estado, para que se pudesse criar uma opção de configuração de trocar o Estado, teria que se adotar a estrutura de sistema na qual o aplicativo teria de usar uma conexão com a internet para baixar o arquivo de banco de dados do estado solicitado pelo usuário e isso envolveria não somente uma conexão com a internet mas também a necessidade da existência de um servidor onde seriam armazenados todos os arquivos de cada estado juntamente com algum *webservice* para controlar todas essas requisições de downloads.

6.1.3 Adaptação do aplicativo para o funcionar totalmente online

O foco do aplicativo aqui desenvolvido é o de ser uma ferramenta prática que não exija uma conexão com a internet para funcionar, pois todos os dados dos estabelecimentos estão armazenados em um banco de dados interno. Tanto os processos de consulta aos dados quanto a sua exibição utilizam dos recursos do dispositivo para serem executadas. Sendo assim para um cenário onde tem-se estabelecimentos de um único estado do país, tal técnica poderia funcionar relativamente bem. Porém, em um cenário onde o mesmo projeto fosse adaptado para comportar dados de estabelecimentos de outros estados do Brasil, a técnica utilizada aqui deveria ser abandonada. Provavelmente, esta técnica geraria um tamanho de arquivo que ultrapassaria os limites permitidos pela loja de aplicativos da Google, mesmo tal limite de tamanho ter sido dobrado de 50MB para 100MB com afirmado por Farinaccio (2016), ainda assim a questão do processamento das consultas poderia causar possíveis gargalos.

Para uma adaptação onde se teriam informações de todos os estabelecimentos do país, abandonar-se-ia a ideia de aplicativo off-line. Dever-se-ia adotar uma estrutura onde um servidor online armazenaria os dados dos estabelecimentos disponibilizá-los-ias através de um *webservice*. O aplicativo, através

de conexão com a internet consumiria tal serviço solicitando consultas ao banco de dados armazenado no servidor. Dessa forma o custo computacional para as consultas ficaria a cargo do servidor e o aplicativo seria focado na realização de requisições ao servidor e apresentação dos dados recebidos ao usuário.

6.1.4 Reengenharia da interface gráfica

O projeto desenvolvido tinha como foco principal a criação de um aplicativo para localizar estabelecimentos de saúde dentro do Estado de Santa Catarina. Todos os esforços gastos foram utilizados para atingir este objetivo. E tal objetivo pode ser considerado atingido, porém, o resultado de todo esse trabalho gerou algumas questões que podem ser repensadas em relação a usabilidade do aplicativo. Dessa forma, propõem-se como trabalhos futuros um estudo aprofundado utilizando as mais modernas técnicas de engenharia da usabilidade visando a criação de novas telas, mais intuitivas e de simples de se utilizar.

REFERÊNCIAS

ALVES, Paulo. **A evolução do Android: do Cupcake ao Marshmallow, conheça todas as versões**. Disponível em:

<<http://www.techtudo.com.br/noticias/noticia/2015/10/a-evolucao-do-android-do-cupcake-ao-marshmallow-conheca-todas-as-versoes.html>>. Acesso em: 31 ago. 2016.

ANDROID DEVELOPER. **Dashboards**. Disponível em:

<<https://developer.android.com/about/dashboards/index.html#Platform>>. Acesso em: 20 mai. 2016.

ANDROID OPEN SOURCE PROJECT. **Android Compatibility**. Disponível em:

<<http://source.android.com/compatibility/index.html>>. Acesso em: 30 ago. 2016.

ANDROID OPEN SOURCE PROJECT. **Governance Philosophy**. Disponível em:

<<http://source.android.com/source/index.html#governance-philosophy>>. Acesso em: 30 ago. 2016.

ANDROID OPEN SOURCE PROJECT. **Life of a Patch**. Disponível em:

<<http://source.android.com/source/life-of-a-patch.html>>. Acesso em: 30 ago. 2016.

ANDROID OPEN SOURCE PROJECT. **Project Roles**. Disponível em:

<<http://source.android.com/source/roles.html>>. Acesso em: 30 ago. 2016.

ANDROID STUDIO. **User Guide**. Disponível em:

<<https://developer.android.com/studio/projects/index.html>>. Acesso em: 21 set. 2016.

ANDROID. **Codenames, Tags, and Build Numbers**. 2016a. Disponível em:

<<http://source.android.com/source/build-numbers.html#platform-code-names-versions-api-levels-and-ndk-releases>>. Acesso em: 30 ago. 2016.

ANDROID. **Platform Architecture**. 2016b. Disponível em:

<<https://developer.android.com/guide/platform/index.html>>. Acesso em: 1 set. 2016.

BRASIL. MINISTRO DE ESTADO DA SAÚDE. Portaria nº 1.646, de 02 de outubro de 2015. **Diário Oficial da União**, Brasília, 2 out. 2015, v. , n. 190, p. 669-670.

BRASIL. MS-SAS - Secretaria de Atenção a Saúde. Portaria nº 376, de 03 de outubro de 2000. **Diário Oficial da União**, Brasília, 3 out. 2000, v. , n. 192-E, p. 19-39.

BRASIL. MS-SAS: Secretaria de Assistência a Saúde . Portaria nº 511, de 29 de dezembro de 2000. **Diário Oficial da União**, Brasília, 4 jan. 2001, v. , n. 3-E, p. 11-20.

BRASIL. MS-SAS - Secretaria de Atenção a Saúde. Portaria nº 682, de 10 de dezembro de 2010. **Diário Oficial da União**, Brasília, 10 dez. 2010, v. , n. 237, p. 52-53.

BUENO, Juliano Martins. **Ciro Radiologia Odontológica**. 2016. Disponível em: <<https://play.google.com/store/apps/details?id=com.embarcadero.ciro>>. Acesso em: 06 dez. 2016.

CLINICA PRIVADA DE ESPECIALIDADES VILLA MARÍA. **Clinica Especialidades**. Disponível em: <<https://play.google.com/store/apps/details?id=clinicaespecialidades.myapp.name>>. Acesso em: 25 out. 2015.

CNES. **CONSULTA LEGISLAÇÃO**. 2016. Disponível em: <<http://cnes.datasus.gov.br/pages/legislacoes.jsp>>. Acesso em: 06 dez. 2016.

DATASUS. **CNES - Cadastro Nacional de Estabelecimentos de Saúde**. Disponível em: <<http://datasus.saude.gov.br/sistemas-e-aplicativos/cadastros-nacionais/cnes>>. Acesso em: 6 set. 2016.

DATASUS. **Consultas - Limites de Competências**. 2015a. Disponível em: <<http://200.214.44.204/transmissor-web/pages/consultaLimiteCompetencia.jsp?cmbNomeSistema=CNES>>. Acesso em: 21 nov. 2015.

DATASUS. **Estabelecimento de saúde**: Portal CNESnet. 2015b. Disponível em: <http://cnes2.datasus.gov.br/Exibe_Ficha_Estabelecimento.asp?VCo_Unidade=4205403157245&VEstado=42&VCo;>. Acesso em: 23 maio 2015.

DATASUS. **SCNES: Sistema de Cadastro Nacional de Estabelecimentos de Saúde - Manual de Instalação e Administração**. 3 ed. Rio de Janeiro: MS/DATASUS/CTI, 2009a.

DATASUS. **SCNES: Sistema de Cadastro Nacional de Estabelecimentos de Saúde - Manual de Operação**. 3 ed. Rio de Janeiro: MS/DATASUS/CTI, 2009b.

DEITEL, Paul; DEITEL, Harvey; WALD, Alexander. **Android 6 para programadores : uma abordagem baseada em aplicativos**. 3 ed. Porto Alegre: Bookman, 2016.

DEVDECISION CONSULTORIA EM TECNOLOGIA. **SIPS - Guia Hospitais**. Disponível em: <<https://play.google.com/store/apps/details?id=com.sipsv2>>. Acesso em: 24 mai. 2015.

EXAME.COM. **Vendas mundiais de smartphones crescem 23% em 2014**. Disponível em: <<http://exame.abril.com.br/tecnologia/noticias/vendas-mundiais-de-smartphones-crescem-23-em-2014>>. Acesso em: 23 mai. 2015.

FARINACCIO, Rafael. **Limite de tamanho de apps na Google Play Store sobe de 50 MB para 100 MB**. Disponível em: <<http://www.tecmundo.com.br/google-play/87155-limite-tamanho-apps-google-play-store-de-50-mb-100-mb.htm>>. Acesso em: 4 out. 2016.

FREEMAN, Eric; FREEMAN, Elisabeth; SIERRA, Kathy; et al. **Use a Cabeça**:

Padrões e Projetos. 2 ed. Rio de Janeiro: Alta Books, 2009.

GLOBALDEV - SISTEMAS WEB E APLICATIVOS MOBILE. **Clinica Santa Cecilia**. Disponível em: <<https://play.google.com/store/apps/details?id=br.clinicasantacecilia>>. Acesso em: 25 out. 2015.

GOOGLE DESIGN. **Material Design Guidelines**. Disponível em: <<https://www.google.com/design/spec/material-design/introduction.html>>. Acesso em: 31 mai. 2016.

HAMANN, Renan. **Android: entenda as diferenças entre o ART e o Dalvik**. Disponível em: <<http://www.tecmundo.com.br/android/54387-android-entenda-diferencas-entre-art-o-dalvik.htm>>. Acesso em: 1 set. 2016.

HEDLEY, Jonathan. **JSoup: Java HTML Parser**. Disponível em: <<https://jsoup.org/>>. Acesso em: 26 out. 2015.

JUNIOR, Alessandro A.. **Android Marshmallow: veja quais apps estão consumindo memória RAM**. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2015/12/android-marshmallow-veja-quais-apps-estao-consumindo-memoria-ram.html>>. Acesso em: 12 out. 2016.

LG. **LG NEXUS 5: ESPECIFICAÇÃO TÉCNICA**. Disponível em: <<http://www.lg.com/br/celulares/lg-D821>>. Acesso em: 12 out. 2016.

LUZZI, Luciano. **Android – Persistência de Dados usando SQLite**. Disponível em: <<http://www.mobiltec.com.br/blog/index.php/android-persistencia-de-dados-usando-sqlite>>. Acesso em: 29 out. 2015.

MEDSQUARE. **medsquare**. Disponível em: <<http://www.medsquare.com.br>>. Acesso em: 25 out. 2015.

MERCATO, Mattia. **A doce história do Android**. Disponível em: <<http://www.androidpit.com.br/historia-do-android>>. Acesso em: 31 ago. 2016.

MEYER, Maximiliano. **A história do Android**. Disponível em: <<https://www.oficinadanet.com.br/post/13939-a-historia-do-android>>. Acesso em: 31 ago. 2016.

MINISTÉRIO DA SAÚDE. **MANUAL TÉCNICO DO CADASTRO NACIONAL DE ESTABELECIMENTOS DE SAÚDE - VERSÃO 2**. 1 ed. Brasília: Ministério da Saúde, 2006.

NAÍMA SALEH. Revista Crescer. **Mais de 70% dos pais exageram no uso de smartphones, aponta pesquisa**. 2015. Portal GLOBO.COM. Disponível em: <<http://revistacrescer.globo.com/Familia/Rotina/noticia/2015/08/mais-de-70-dos-pais-exageram-no-uso-d>>. Acesso em: 06 dez. 2016.

OLIVEIRA, Caio. **DBDesigner Fork**. Disponível em: <<http://www.caiooliveira.com.br/?p=66>>. Acesso em: 30 mai. 2015.

PORTAL UNIMED LTDA. **Guia Médico Nacional Unimed para smartphone.**

Disponível em:

<<https://play.google.com/store/apps/details?id=com.unimed.guiamdico>>. Acesso em: 23 mai. 2015.

SARINHO, Victor Travassos; CAMPOS, Luan Rios. SAMob - Sistema Móvel de Geolocalização e Geoprocessamento para Locais de Atendimento em Saúde.

Journal of Health Informatics. São Paulo; v. 6, n. 3, set. 2014. p. 96-101.

Disponível em: <<http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/view/304>>. Acesso em: 18 out. 2015.

SCHMIDT, Roberto Hammer. **Linha do Tempo: por dentro da evolução do**

Android. Disponível em: <<http://www.tecmundo.com.br/android/82344-linha-tempo-dentro-evolucao-do-sistema-android.htm>>. Acesso em: 31 ago. 2016.

SOCIAL ADVISORS. **Clínica Bicentenário.** Disponível em:

<https://play.google.com/store/apps/details?id=biz.app4mobile.app_4bb5870e058a4ae9b067d271d142c996>. Acesso em: 25 out. 2015.

TABORDA, Sérgio Manuel Marcos. **DAO.** Disponível em:

<<https://sergiotaborda.wordpress.com/desenvolvimento-de-software/java/patterns/dao/>>. Acesso em: 30 mai. 2015.

TAVARES, Alberto Trindade; LÓSCIO, Bernadette Farias. Processamento de consultas na Web de Dados: uma abordagem para busca de fontes de dados relevantes. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS. 28., 2013, Recife.

Short Papers... Recife: SBC, 2013. Disponível em:

<<http://sbbd2013.cin.ufpe.br/Proceedings/artigos/pdfs/SBBD-Proceedings-Complete.pdf>>. Acesso em: 25 out. 2015.

TAVARES, Alberto Trindade; LÓSCIO, Bernadette Farias; OLIVEIRA, Hélio Rodrigues de. Buscando Fontes de Dados Relevantes para Aplicações Linked Data. In: SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E WEB. 18., 2012, São Paulo.

Anais IX Workshop de Trabalhos de Iniciação Científica... São Paulo:

SBC, 2012. Disponível em:

<http://sws2012.ime.usp.br/download/WebMedia2012_Anais.pdf>. Acesso em: 25 out. 2015.

VOXAGE SERVIÇOS INTERATIVOS. **MEATENDE.** Disponível em:

<<http://www.voxage.com.br/meatende/>>. Acesso em: 06 dez. 2016.

WINTIMEIPC. **MedSquare.** Disponível em:

<https://play.google.com/store/apps/details?id=br.com.medsquare.medsquare&hl=pt_BR>. Acesso em: 24 mai. 2015.

XAVIER, Renan. **Clinica Santa Cecilia.** Disponível em:

<<https://play.google.com/store/apps/details?id=br.clinicasantacecilia>>. Acesso em: 25 out. 2015.

APÊNDICES

APÊNDICE A - ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE

Especificação de Requisitos do Aplicativo CadESaúde

Projeto: CadESaúde

Versão	Autor	Data	Ação
1.0	Jucemar Dimon	15/12/2015	Estabelecimento de Requisitos

Conteúdo:

1. Introdução
2. Visão Geral
3. Requisitos de Software

Introdução:

Objetivo: Desenvolvimento de um aplicativo para a plataforma Android que permita ao usuário portando um smartphone com o programa devidamente instalado localizar um estabelecimento de saúde de forma mais eficiente.

Descrição: CadESaúde é um aplicativo para smartphone que será desenvolvido com o intuito de auxiliar o cidadão, morador ou turista, do Estado de Santa Catarina a localizar um estabelecimento que presta atendimento em saúde seja ele privado ou público. Entende-se por estabelecimento que presta atendimento em saúde como: clínicas ou consultórios médicos, clínicas ou consultórios odontológicos, policlínicas, hospitais, entre outros. Portando de um smartphone com o aplicativo instalado e o serviço de localização ativo, será possível visualizar uma lista de todos os estabelecimentos de saúde próximos dentro de um raio de circunferência que deve ser pré-definido pelo usuário nas configurações do aplicativo. Em posse dessa lista de locais o usuário terá a opção de utilizar diversos filtros para auxiliar na visualização do que procura no momento. Um filtro pode ser aplicado um em sobreposição a outro para aumentar a eficácia da busca. As opções filtros vão de tipo de estabelecimento, passando por serviços oferecidos até os tipos de convênios aceitos. Além da busca por localização geográfica que auxiliará o usuário a encontrar estabelecimentos independente da cidade onde estiver dentro do estado de Santa Catarina, há ainda a possibilidade de encontrar estabelecimentos dentro das cidades pré-definidas nas configurações do aplicativo. Nessa categoria de busca será possível realizar buscas tabuladas por tipos de estabelecimentos, por tipos de especialidades médicas existentes nestes além da possibilidade de buscas por nome. O aplicativo será desenvolvido e sua versão final e utilizável será publicada na loja de aplicativos da Google.

Visão Geral:

Paradigma do Programa: Programa Orientado a Objetos.

Premissas de Desenvolvimento: O aplicativo será desenvolvido na linguagem Java utilizando a Android API 4.0.3 – API Level 15 - e deverá ser compatível com aparelhos smartphones que utilizem sistema operacional Android em suas versões 4.0.3 - Ice Cream Sandwich ou superiores. A interface gráfica com o usuário será desenvolvida utilizando os artefatos mais modernos nativos da versão mais atualizada da API do Android que no momento do desenvolvimento deste projeto seria a 6.0 juntamente com os componentes visuais fornecidos pelo padrão visual Material Design com sua biblioteca de compatibilidade com versões antigas.

Requisitos de Software:

Requisitos Funcionais:

Requisito funcional 1 – Definir raio de distância: A tela de configurações do aplicativo deverá exibir ao usuário a opção de definir um raio de distância de sua preferência. Tal informação será utilizada na funcionalidade de busca por estabelecimentos através de localização geográfica.

Requisito funcional 2 – Definir municípios de preferência: A tela de configurações do aplicativo deverá exibir ao usuário uma lista de municípios de Santa Catarina na qual este poderá selecionar o seu município de residência e outros adjacentes de seu interesse. Tais informações serão obrigatórias para o funcionamento do aplicativo pois através delas serão possíveis realizar as buscas tabuladas onde os estabelecimentos resultantes serão aqueles instalados nos municípios pré-definidos no aplicativo.

Requisito funcional 3 – Visualizar estabelecimentos favoritos: O aplicativo deve permitir ao usuário visualizar uma lista contendo todos os estabelecimentos de saúde que foram marcados como favoritos em uma situação anterior.

Requisito funcional 4 – Buscar estabelecimentos pelas redondezas: O aplicativo deverá permitir ao seu utilizador usar a sua posição geográfica atual, adquirida através do GPS do smartphone, para visualizar os estabelecimentos nas redondezas até um limite de distância pré-definido anteriormente pelo usuário no menu de configurações do aplicativo.

Requisito funcional 5 – Filtrar resultados de buscas por estabelecimentos: O aplicativo deverá permitir que filtros possam ser aplicados aos resultados das buscas por estabelecimentos nas localidades com o intuito de facilitar a identificação de uma característica específica de interesse do usuário. Deve haver a possibilidade de aplicação de filtro sobre filtro para tornar o processo mais eficiente. Os filtros devem ter relação com os dados disponíveis no banco de dados referenciando cada estabelecimento. Os filtros a serem disponibilizados aos usuários serão:

- Tipos de convênios;
- Tipos de atendimentos;
- Tipos de profissionais;
- Tipos de serviços;
- Tipos de estabelecimentos.

Requisito funcional 6 – Visualizar os detalhes de um estabelecimento: Deverá ser possível ao usuário visualizar os dados detalhados do estabelecimento encontrado. Entende-se por dados detalhados todas as informações relacionadas a este estabelecimento que estão armazenadas no banco de dados do aplicativo. A principais informações que deverão ser mostradas na tela são:

- Informações empresariais: nome fantasia, razão social;
- Informações sobre a localização onde está instalado o negócio;
- Informações sobre o tipo de estabelecimento: se ele é uma clínica, um hospital, uma policlínica, entre

- outros;
- Telefones de contato;
- Informações sobre os profissionais que prestam atendimento no local;
- Informações sobre os serviços oferecidos pela empresa;
- Informações sobre os tipos de atendimento realizado no local: se faz atendimento de urgência, se faz internação do paciente, se realiza exames no estabelecimento, entre outros;
- Informações sobre os tipos de convênios que aceita: SUS, particular, privado, entre outros.

Requisito funcional 7 – Visualizar mapa com a localização geográfica do estabelecimento: O aplicativo deve permitir ao usuário a possibilidade de visualização de um mapa contendo a localização geográfica do estabelecimento que está sendo visualizado em detalhes. Tal mapa não será fornecido junto com o aplicativo. Este somente fará a comunicação com o sistema operacional Android e repassará alguns dados que possibilitem ao aplicativo que foi definido pelo usuário como aplicativo padrão para visualização de mapas para que este desempenhe suas tarefas de marcar o estabelecimento no mapa, definir rotas até este ponto, entre outros.

Requisito funcional 8 – Definir estabelecimento como favorito: O aplicativo deve permitir ao usuário optar por marcar ou desmarcar como favorito um estabelecimento de saúde que esteja sendo visualizado de forma detalhada na tela do celular.

Requisito funcional 9 – Realizar uma ligação telefônica para um estabelecimento: O aplicativo deve permitir ao usuário optar por fazer uma ligação telefônica para o estabelecimento que está tendo seus dados detalhados visualizados. O aplicativo não fará por conta própria a ligação, ele simplesmente solicitará ao sistema operacional Android que abra o aplicativo padrão de ligações telefônicas e faça uma ligação para o número de telefone e a partir deste a ação será concluída.

Requisito funcional 10 – Buscar estabelecimentos de forma generalizada: O aplicativo deve permitir ao usuário buscar estabelecimentos de acordo com alguma característica específica ou comum entre os estabelecimentos. As possibilidades de busca serão:

- Busca por estabelecimentos que trabalham com tipo de convenio específico. Neste caso o usuário deverá escolher uma dentre uma relação pré-definida de tipos de convênio;
- Busca por estabelecimentos que possuam uma forma de atendimento específica. Nesta situação o usuário deverá escolher uma dentre as possíveis disponíveis para realizar sua busca;
- Busca por estabelecimentos que possuam em seu quadro de funcionários um profissional com um nome equivalente ao informado pelo usuário;
- Busca por estabelecimentos que possuam um profissional do tipo escolhido pelo usuário de uma lista de possibilidades;
- Busca por estabelecimentos que possuam em seu nome fantasia ou razão social um termo a ser informado pelo usuário;
- Busca por estabelecimentos que possuam um tipo de serviço listado e escolhido pelo usuário do aplicativo durante o processo de busca.

Os resultados das buscas realizadas nesta funcionalidade são limitados de acordo com o município de instalação do estabelecimento, dessa forma somente aparecerão nos resultados as empresas localizadas dentro dos municípios pré-definidos pelo usuário na tela de configuração da aplicação.

Requisito funcional 11 – Fornecer ajuda ao usuário do aplicativo: O aplicativo deverá fornecer ao usuário um menu onde ele poderá tirar algumas dúvidas em relação aos termos técnicos utilizados nas nomenclaturas das áreas da saúde e medicina.

Requisitos Não-Funcionais:

Requisito não-funcional 1 – Sistema de localização: O smartphone no qual será instalado o aplicativo deverá possuir um sistema de localização baseado em GPS e este deverá fornecer a localização geográfica do aparelho de forma rápida e precisa com risco de comprometer o desempenho final do aplicativo caso este não atenda aos requisitos definidos.

Requisito não-funcional 2 – Independência de conexão com a internet: O aplicativo desenvolvido não depende

da conexão com a internet para executar suas funções básicas planejadas, porém algumas funções adicionais possíveis de serem realizadas pelo mesmo como a instalação e atualização do aplicativo necessitam que o aparelho no qual ele esteja instalado ou irá ser instalado pela primeira vez, possua uma conexão ativa com a internet pois é através dela que os avisos sobre novas versões lançadas serão enviados.

Requisito não-funcional 3 – Aplicativo de visualização de mapas: O smartphone no qual o CadESaúde será instalado deverá possuir um aplicativo padrão para visualização de mapas. Este aplicativo poderá exigir ou não uma conexão com a internet, não existe restrição quanto a isso.

Requisito não-funcional 4 – Tipo de dispositivo: CadESaúde é compatível somente com smartphones, logo, tablets, TVs, wearables e autos, entre outros, não serão compatíveis. Ele funcionará somente em modo retrato (*portrait*) será instalado deverá possuir um aplicativo padrão para visualização de mapas. Este aplicativo poderá exigir ou não uma conexão com a internet, não existe restrição quanto a isso.

APÊNDICE B - CODIGO FONTE DO PROGRAMA DE CAPTURA DE DADOS

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.entidades;
2
3 import java.io.Serializable;
4
5 /**
6  * @author Jucemar
7  */
8 public class AtendimentoPrestado implements Serializable {
9
10     private String tipoAtendimento;
11     private String convenio;
12
13     public AtendimentoPrestado() {
14     }
15
16     public String getTipoAtendimento() {
17         return tipoAtendimento;
18     }
19
20     public void setTipoAtendimento(String tipoAtendimento) {
21         this.tipoAtendimento = tipoAtendimento;
22     }
23
24     public String getConvenio() {
25         return convenio;
26     }
27
28     public void setConvenio(String convenio) {
29         this.convenio = convenio;
30     }
31 }
```


E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/Estabelecimento.java

```
1 package br.com._jucemar_dimon._jwebcrawlercnes.entidades;
2
3 import java.io.Serializable;
4 import java.util.ArrayList;
5
6 /**
7  * @author Jucemar
8  */
9 public class Estabelecimento implements Serializable {
10
11     private int cnes;
12
13     private String razaoSocial;
14     private String nomeFantasia;
15     private String logradouro;
16     private String numero;
17     private String complemento;
18     private String bairro;
19     private String cep;
20     private String municipio;
21     private String estado;
22     private String telefone;
23     private String tipoEstabelecimento;
24     private String codigoMunicipio;
25     private Double latitude;
26     private Double longitude;
27     private ArrayList<Profissional> profissionaisVinculados;
28     private ArrayList<ServicoClassificacao> servicosOferecidos;
29     private ArrayList<AtendimentoPrestado> atendimentosPrestados;
30
31     public Estabelecimento() {
32     ..
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/Estabelecimento.java

```
32 this.profissionaisVinculados = new ArrayList<Profissional>();
33 this.servicosOferecidos = new ArrayList<ServicoClassificacao>();
34 this.atendimentosPrestados = new ArrayList<AtendimentoPrestado>();
35 }
36
37 public Estabelecimento(int cnes, String razaoSocial, String nomeFantasia, String nomeLogradouro, String numero,
String complemento, String bairro, String cep, String municipio, String codigoMunicipio, String estado, String telefone,
String tipoEstabelecimento, Double latitude, Double longitude) {
38     this.cnes = cnes;
39     this.razaoSocial = razaoSocial;
40     this.nomeFantasia = nomeFantasia;
41     this.logradouro = nomeLogradouro;
42     this.numero = numero;
43     this.complemento = complemento;
44     this.bairro = bairro;
45     this.cep = cep; this.municipio =
municipio; this.codigoMunicipio =
codigoMunicipio; this.estado = estado;
46     this.telefone = telefone;
47     this.tipoEstabelecimento = tipoEstabelecimento;
48     this.latitude = latitude;
49     this.longitude = longitude;
50 }
51
52
53
54 public String getCodigoMunicipio() {
55     return codigoMunicipio;
56 }
57
58 public void setCodigoMunicipio(String codigoMunicipio) {
59     this.codigoMunicipio = codigoMunicipio;
60 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/Estabelecimento.java

```
61     }
62
63     public ArrayList<ServicoClassificacao> getServicosOferecidos() {
64         return servicosOferecidos;
65     }
66
67     public void setServicosOferecidos(ArrayList<ServicoClassificacao> ServicosOferecidos) {
68         this.servicosOferecidos = ServicosOferecidos;
69     }
70
71     public ArrayList<Profissional> getProfissionaisVinculados() {
72         return profissionaisVinculados;
73     }
74
75     public void setProfissionaisVinculados(ArrayList<Profissional> profissionaisVinculados) {
76         this.profissionaisVinculados = profissionaisVinculados;
77     }
78
79     public int getCnes() {
80         return cnes;
81     }
82
83     public void setCnes(int cnes) {
84         this.cnes = cnes;
85     }
86
87     public String getLogradouro() {
88         return logradouro;
89     }
90
91     public void setLogradouro(String nomeLogradouro) {
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/Estabelecimento.java

```
92     this.logradouro = nomeLogradouro;
93 }
94
95 public String getEstado() {
96     return estado;
97 }
98
99 public void setEstado(String estado) {
100     this.estado = estado;
101 }
102
103 public Double getLatitude() {
104     return latitude;
105 }
106
107 public void setLatitude(Double latitude) {
108     this.latitude = latitude;
109 }
110
111 public Double getLongitude() {
112     return longitude;
113 }
114
115 public void setLongitude(Double longitude) {
116     this.longitude = longitude;
117 }
118
119 public String getRazaoSocial() {
120     return razaoSocial;
121 }
122
123 --
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/Estabelecimento.java

```
123 public void setRazaoSocial(String razaoSocial) {
124     this.razaoSocial = razaoSocial;
125 }
126
127 public String getNomeFantasia() {
128     return nomeFantasia;
129 }
130
131 public void setNomeFantasia(String nomeFantasia) {
132     this.nomeFantasia = nomeFantasia;
133 }
134
135 public String getNumero() {
136     return numero;
137 }
138
139 public void setNumero(String numero) {
140     this.numero = numero;
141 }
142
143 public String getComplemento() {
144     return complemento;
145 }
146
147 public void setComplemento(String complemento) {
148     this.complemento = complemento;
149 }
150
151 public String getBairro() {
152     return bairro;
153 }
154
```



```
154
155 public void setBairro(String bairro) {
156     this.bairro = bairro;
157 }
158
159 public String getCep() {
160     return cep;
161 }
162
163 public void setCep(String cep) {
164     this.cep = cep;
165 }
166
167 public String getMunicipio() {
168     return municipio;
169 }
170
171 public void setMunicipio(String municipio) {
172     this.municipio = municipio;
173 }
174
175 public String getTelefone() {
176     return telefone;
177 }
178
179 public void setTelefone(String telefone) {
180     this.telefone = telefone;
181 }
182
183 public String getTipoEstabelecimento() {
184     return tipoEstabelecimento;
185 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/Estabelecimento.java

```
185     }
186
187     public void setTipoEstabelecimento(String tipoEstabelecimento) {
188         this.tipoEstabelecimento = tipoEstabelecimento;
189     }
190
191     public ArrayList<AtendimentoPrestado> getAtendimentosPrestados() {
192         return atendimentosPrestados;
193     }
194
195     public void setAtendimentosPrestados(ArrayList<AtendimentoPrestado> atendimentosPrestados) {
196         this.atendimentosPrestados = atendimentosPrestados;
197     }
198
199 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/EstabelecimentoResumido.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.entidades;
2
3 import java.io.Serializable;
4
5 /**
6  * @author Jucemar
7  */
8 public class EstabelecimentoResumido implements Serializable {
9
10     private String nome;
11     private String cnes;
12     private String codigoMunicipio;
13     private String idConsulta;
14     private String linkDaPagina;
15
16     public EstabelecimentoResumido(String nome, String cnes, String codigoMunicipio, String linkDaPagina) {
17         this.nome = nome;
18         this.cnes = cnes;
19         this.codigoMunicipio = codigoMunicipio;
20         this.idConsulta = codigoMunicipio + cnes;
21         this.linkDaPagina = linkDaPagina;
22     }
23
24     public String getCodigoMunicipio() {
25         return codigoMunicipio;
26     }
27
28     public void setCodigoMunicipio(String codigoMunicipio) {
29         this.codigoMunicipio = codigoMunicipio;
30     }
31 }
```

```
32 public String getIdConsulta() {
33     return idConsulta;
34 }
35
36 public void setIdConsulta(String idConsulta) {
37     this.idConsulta = idConsulta;
38 }
39
40 public String getNome() {
41     return nome;
42 }
43
44 public void setNome(String nome) {
45     this.nome = nome;
46 }
47
48 public String getCnes() {
49     return cnes;
50 }
51
52 public void setCnes(String cnes) {
53     this.cnes = cnes;
54 }
55
56 public String getLinkDaPagina() {
57     return linkDaPagina;
58 }
59
60 public void setLinkDaPagina(String linkDaPagina) {
61     this.linkDaPagina = linkDaPagina;
62 }
```

63

64 }

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/Municipio.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.entidades;
2
3 import java.io.Serializable;
4 import java.util.ArrayList;
5
6 /**
7  * @author Jucemar
8  */
9 public class Municipio implements Serializable {
10
11     private String codigo;
12     private String nome;
13     private int     qtdeEstabelecimentos;    String
14     private LinkDaPagina;    ArrayList<Estabelecimento>
15     private estabelecimentos;
16
17     public Municipio(String codigo, String nome, int qtdeEstabelecimentos, String linkDaPagina) {
18         this.codigo = codigo;
19         this.nome = nome;
20         this.qtdeEstabelecimentos = qtdeEstabelecimentos;
21         this.linkDaPagina = linkDaPagina;
22         estabelecimentos = null;
23     }
24
25     public ArrayList<Estabelecimento> getEstabelecimentos() {
26         return estabelecimentos;
27     }
28
29     public void setEstabelecimentos(ArrayList<Estabelecimento> estabelecimentos) {
30         this.estabelecimentos = estabelecimentos;
31     }
32 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/Municipio.java

```
32
33     public String getCodigo() {
34         return codigo;
35     }
36
37     public void setCodigo(String codigo) {
38         this.codigo = codigo;
39     }
40
41     public String getNome() {
42         return nome;
43     }
44
45     public void setNome(String nome) {
46         this.nome = nome;
47     }
48
49     public int getQtdeEstabelecimentos() {
50         return qtdeEstabelecimentos;
51     }
52
53     public void setQtdeEstabelecimentos(int qtdeEstabelecimentos) {
54         this.qtdeEstabelecimentos = qtdeEstabelecimentos;
55     }
56
57     public String getLinkDaPagina() {
58         return linkDaPagina;
59     }
60
61     public void setLinkDaPagina(String linkDaPagina) {
62         this.linkDaPagina = linkDaPagina;
63     }
64 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/Municipio.java

63 }

64

65 }

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/Profissional.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.entidades;
2
3 import java.io.Serializable;
4
5 /**
6  * @author Jucemar
7  */
8 public class Profissional implements Serializable {
9
10     private String nome;
11     private String cbo;
12     private String cboDescricao;
13
14     public Profissional() {
15     }
16
17     public String getNome() {
18         return nome;
19     }
20
21     public void setNome(String nome) {
22         this.nome = nome;
23     }
24
25     public String getCbo() {
26         return cbo;
27     }
28
29     public void setCbo(String cbo) {
30         this.cbo = cbo;
31     }
32 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/Profissional.java

```
32
33     public String getCboDescricaoO() {
34         return cboDescricao;
35     }
36
37     public void setCboDescricao(String cboDescricao) {
38         this.cboDescricao = cboDescricao;
39     }
40
41 }
```

E:/Projetos Netbeans/JWebCrawlerCNETS/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/ServicoClassificacao.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.entidades;
2
3 import java.io.Serializable;
4
5 /**
6  * @author Jucemar
7  */
8 public class ServicoClassificacao implements Serializable {
9
10     private int servico;
11     private int classificacao;
12
13     public ServicoClassificacao(int servico, int classificacao) {
14         this.servico = servico;
15         this.classificacao = classificacao;
16     }
17
18     public ServicoClassificacao() {
19
20     }
21
22     public int getServico() {
23         return servico;
24     }
25
26     public void setServico(int servico) {
27         this.servico = servico;
28     }
29
30     public int getClassificacao() {
31         return classificacao;
32     }
33 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/entidades/ServicoClassificacao.java

```
32     }
33
34     public void setClassificacao(int classificacao) {
35         this.classificacao = classificacao;
36     }
37
38 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/AtendimentoPrestadoDAO.java

```

1 package br.com.jucemar_dimon.jwebcrawlercenes.principal;
2
3 import br.com.jucemar_dimon.jwebcrawlercenes.entidades.AtendimentoPrestado;
4 import java.sql.Connection;
5 import java.sql.DriverManager;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10
11 /**
12  * @author Jucemar
13  */
14 public class AtendimentoPrestadoDAO {
15
16     private Connection c = null;
17     private Statement stmt = null;
18
19     public AtendimentoPrestadoDAO() {
20         try {
21             Class.forName("org.sqlite.JDBC");
22             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
23
24             System.out.println("Conexão com o banco de dados estabelecida com sucesso");
25         }
26     }
27
28     String tabelaAtendimento = "CREATE TABLE IF NOT EXISTS atendimento_estabelecimento "
29         + "(id INTEGER PRIMARY KEY AUTOINCREMENT,"
30         + " atendimento TEXT NOT NULL,"
31         + " convenio TEXT NOT NULL,"

```

```

32 + " estabelecimento_cnes INTEGER NOT NULL,"
33 + " FOREIGN KEY(estabelecimento_cnes)"
34 + " REFERENCES estabelecimento(cnes));";
35
36 String index1 = "CREATE INDEX IF NOT EXISTS atendimento_estabelecimento_PKIndex ON
atendimento_estabelecimento(id);";
37 String index2 = "CREATE INDEX IF NOT EXISTS atendimento_estabelecimento_FKIndex ON
atendimento_estabelecimento (estabelecimento_cnes);";
38
39 stmt = c.createStatement();
40 stmt.executeUpdate(tabelaAtendimento);
41 stmt.executeUpdate(index1);
42 stmt.executeUpdate(index2);
43 stmt.close();
44 c.close();
45 } catch (Exception e) {
46 System.err.println(e.getClass().getName() + ": " + e.getMessage());
47 }
48 System.out.println("Tabela de Atendimento_prestado criada com sucesso");
49 } catch
    (ClassNotFoundException ex)
50 {
    Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
51 } catch (SQLException ex) {
52     Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
53 }
54 }
55
56 public void inserirAtendimento(int cnesEstabelecimento, AtendimentoPrestado atendimento) {
57
58     try {
59         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
60         this.stmt = this.c.createStatement();

```

```
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
    try {
        String sql = "INSERT INTO atendimento_estabelecimento("
            + "atendimento, convenio, estabelecimento_cnes) "
            + "VALUES("
            + atendimento.getTipoAtendimento() + ", ",
            + atendimento.getConvenio() + ", ",
            + cnesEstabelecimento
            + "); ";
        System.out.println(sql);
        this.stmt.executeUpdate(sql);

        } catch (SQLException e) {
            e.printStackTrace();
        }

        stmt.close();
        c.close();
        catch (SQLException ex) {
            Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.principal;
2
3 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.AtendimentoPrestado;
4 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.Estabelecimento;
5 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.EstabelecimentoResumido;
6 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.Municipio;
7 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.Profissional;
8 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.ServicoClassificacao;
9 import java.io.IOException;
10 import java.util.ArrayList;
11 import java.util.logging.Level;
12 import java.util.logging.Logger;
13 import org.jsoup.Connection;
14 import org.jsoup.Connection.Response;
15 import org.jsoup.Jsoup;
16 import org.jsoup.nodes.Document;
17 import org.jsoup.select.Elements;
18 /**
19  * Classe responsável pela extração dos dados das páginas html do portal do
20  * CNEsnet
21  *
22  * @author Jucemar
23  * @version 1.0
24  */
25 public class ConversorHtmlParaJava {
26
27     private static final String FUNCAO_PROFFSSIONAIS_ESTABELECIMENTO = "Mod_Profissional.asp?VCo_Unidade=";
28     private static final String FUNCAO_SERVICO_CLASSIFICACAO = "Mod_Conj_Informacoes.asp?VCo_Unidade=";
29     private static final String FUNCAO_ATENDIMENTOS_PRESTADOS = "Mod_Bas_Atendimento.asp?VCo_Unidade=";
30     private static final String URL_PRINCIPAL = "http://cnes2.datasus.gov.br/";
31 }
```


E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
32 private static final String FUNCAO_ESTABELECIAMENTOS_MUNICIPIOS =
"Lista_Tot_Es_Municipio.asp?Estado=42&NomeEstado=SANTA%20CATARINA";
33 private String googleApiKey = "AIzaSyB4EXy3SiTgmWrGtBIHHSz8rQmTghCO-Yg";
34 private ArrayList<String> estabelecimentosFisicos;
35 int estabelecimentosValidos = 0;
36 int coordLocalizadasOSM = 0;
37 int coordLocalizadasGM = 0;
38
39 /**
40  * Simples construtor padrão
41  */
42 public ConversorHtmlParaJava() { estabelecimentosFisicos = new
43     ArrayList<>(); estabelecimentosFisicos.add("CENTRAL
44     NOTIFICACAO, CAPTACAO E DISTRIB
45     estabelecimentosFisicos.add("CENTRO DE ATENCAO HEMOTERAPIA E OU HEMATOLOGICA");
46     estabelecimentosFisicos.add("CENTRO DE APOIO A SAUDE DA FAMILIA");
47     estabelecimentosFisicos.add("CENTRO DE ATENCAO PSICOSSOCIAL");
48     estabelecimentosFisicos.add("CENTRO DE PARTO NORMAL - ISOLADO");
49     estabelecimentosFisicos.add("CENTRO DE SAUDE/UNIDADE BASICA");
50     estabelecimentosFisicos.add("CLINICA/CENTRO DE ESPECIALIDADE");
51     estabelecimentosFisicos.add("CONSULTORIO
52     estabelecimentosFisicos.add("HOSPITAL
53     estabelecimentosFisicos.add("HOSPITAL
54     estabelecimentosFisicos.add("HOSPITAL/DIA
55     estabelecimentosFisicos.add("LABORATORIO CENTRAL DE SAUDE PUBLICA LACEN");
56     estabelecimentosFisicos.add("LABORATORIO DE SAUDE PUBLICA");
57     estabelecimentosFisicos.add("POLICLINICA");
58     estabelecimentosFisicos.add("POSTO DE SAUDE");
59     estabelecimentosFisicos.add("PRONTO ATENDIMENTO");
60     estabelecimentosFisicos.add("PRONTO SOCORRO ESPECIALIZADO");
61     estabelecimentosFisicos.add("PRONTO SOCORRO GERAL");
62 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
62     estabelecimentosFisicos.add("UNIDADE DE APOIO DIAGNOSE E TERAPIA (SADT ISOLADO)");
63     estabelecimentosFisicos.add("UNIDADE MISTA");
64 }
65
66 /**
67  * Método que retorna uma Collection do tipo ArrayList que armazenam dados
68  * dos municípios como Código IBGE, Nome, Quantidades de estabelecimentos que
69  * possuem e uma String que representa uma url para a página web onde é
70  * possível visualizar os estabelecimentos instalados no município.
71  *
72  * @return ArrayList com objetos do tipo Municipio
73  */
74 public ArrayList<Municipio> getMunicipios() {
75     int statusCode = -1;
76     int qtdeTentativasRequisicaoM = 0;
77     int timeOut = 1000;
78     Response respostaServidor = null;
79     ArrayList<Municipio> resultado = new ArrayList<Municipio>();
80     int qtdeTentativasRequisicao = 0;
81     boolean fim = false;
82     while (fim == false) {
83         Document paginaHtmlMunicipios = null;
84         try {
85             respostaServidor = (Response) Jsoup
86                 .connect(URL_PRINCIPAL + FUNCAO_ESTABELECIMENTOS_MUNICIPIOS)
87                 .ignoreContentType(true)
88                 .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
89                 .referrer("http://www.google.com")
90                 .timeout(timeOut)
91                 .followRedirects(true)
92                 .execute();
93     }
```

```

93     statusCode = respostaServidor.statusCode();
94     if (statusCode == 200) {
95         timeOut = 500; fim = true;
96         qtdeTentativasRequisicao = 1;
97         System.out.println("Conexão estabelecida com a página de municípios do estado
98         de Santa
Catarina.");
99         paginaHtmlMunicipios = respostaServidor.parse();
100         Elements tables = paginaHtmlMunicipios.getElementsByTag("table");
101         if (tables.size() < 8) {
102             fim = false;
103             System.out.println("Erro - página de Municípios de Santa Catarina fora do ar. Uma nova
tentativa será realizada");
104         } else {
105             Elements trs = tables.get(4).getElementsByTag("tr");
106             for (int i = 0; i < trs.size(); i++) {
107                 Elements tds = trs.get(i).getElementsByTag("td");
108                 if (tds.size() < 3) {
109                     qtdeTentativasRequisicaoM++;
110                     System.err.println("Erro - QTDE de TDS: " + trs.get(i).select("td").size());
111                     System.err.println(trs.get(i).select("td"));
112                     System.err.println("Erro - Problemas ao carregar município - linha: " + i);
113                     System.err.println("Erro - Tentativa " + qtdeTentativasRequisicaoM);
114                     i--;
115                     if (qtdeTentativasRequisicaoM > 5) {
116                         System.err.println("Impossível ler a linha: " + i + ", ela será descartada.");
117                         qtdeTentativasRequisicaoM = 0;
118                         i++;
119                     }
120                 } else {
121                     resultado.add(new Municipio(

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
122         tds.get(0).text(),
123         tds.get(1).text(),
124         Integer.parseInt(tds.get(2).text()),
125         URL_PRINCIPAL + tds.get(1).getElementsByTag("a").attr("href"))));
126     }
127 }
128
129 }
130
131     } catch (IOException ex) {
132         qtdeTentativasRequisicao++;
133         System.err.println("-----");
134         System.err.println("StatusCode : " + statusCode);
135         System.err.println("Erro - tentativa com erro " + qtdeTentativasRequisicao + " - Tempo limite de
requisição atingido.");
136         System.err.println("Não foi possível conectar a página de municípios do Estado de Santa Catarina.");
137         if (qtdeTentativasRequisicao > 4) {
138             qtdeTentativasRequisicao = 0;
139             timeOut += 100;
140             System.err.println("Atenção! Verifique sua conexão com a internet");
141             System.err.println("Time out aumentado para " + timeOut);
142             System.err.println("Aguardando 5s até a próxima tentativa.");
143             try
144             {
145                 Thread.sleep(5000);
146                 catch (InterruptedException ex1)
147                 {
148                     Logger.getLogger(ConversorHtmlParaJava.class.getName()).log(Level.SEVERE, null, ex1);
149                 }
150             }
151         }
152     }
153 }
```

E:/Projetos Netbeans/JWebCrawlerCNETS/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
152     }
153     return resultado;
154 }
155
156 /**
157  * Método que retorna uma Collection do tipo ArrayList de objetos do tipo
158  * EstabelecimentoResumido que contém informações resumidas dos
159  * estabelecimentos do município
160  *
161  * @param municipio objeto do tipo Municipio com os dados resumidos sobre os
162  * estabelecimentos do municipio
163  * @return ArrayList de objetos do tipo EstabelecimentoResumido
164  */
165 public ArrayList<EstabelecimentoResumido> getEstabelecimentosDoMunicipio(Municipio municipio) {
166     String url = municipio.getLinkDaPagina();
167     ArrayList<EstabelecimentoResumido> resultado = new ArrayList<EstabelecimentoResumido>();
168     int statusCode = -1;
169     boolean fim = false;
170     Response respostaServidor = null;
171     int qtdeTentativasRequisicao = 0;
172     int qtdeTentativasRequisicaoE = 0;
173     int timeout = 1000;
174     while (fim == false) {
175         Document paginaHtmlEstabelecimentosMunicipio = null;
176         try {
177             respostaServidor = Jsoup
178                 .connect(url)
179                 .ignoreContentType(true)
180                 .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
181                 .referrer("http://www.google.com")
182                 .timeout(timeout)
183         }
```

```

183         .followRedirects(true)
184         .execute();
185
186         statusCode = respostaServidor.statusCode();
187         if (statusCode == 200) {
188             timeout = 500;
189             qtdeTentativasRequisicao = 1;
190             fim = true;
191             System.out.println("Conexão estabelecida com a página de estabelecimentos do município de: " +
municipio.getNome());
192             paginaHtmlEstabelecimentosMunicipio = respostaServidor.parse();
193             Elements tables = paginaHtmlEstabelecimentosMunicipio.getElementsByTag("table");
194             if (tables.size() < 6) {
195                 fim = false;
196                 System.out.println("Erro - página de estabelecimentos do município de " + municipio.getNome()
+ " fora do ar. Uma nova tentativa será realizada");
197             } else {
198                 Elements trs = tables.get(4).getElementsByTag("tr");
199                 for (int i = 0; i < trs.size(); i++) {
200                     Elements tds = trs.get(i).getElementsByTag("td");
201                     if (tds.size() < 4) {
202                         qtdeTentativasRequisicaoE++;
203                         System.err.println("Erro - QTDE de TDS: " + tds.size());
204                         System.err.println(trs.get(i).select("td"));
205                         System.err.println("Erro linha: " + i + " - Problemas ao carregar estabelecimento do
município " + municipio.getNome());
206                     }
207                     System.err.println("Erro - Tentativa " + qtdeTentativasRequisicaoE);
208                     i--;
209                     if (qtdeTentativasRequisicaoE > 5) {
210                         System.err.println("Impossível ler a linha: " + i + ", ela será descartada.");
211                         qtdeTentativasRequisicaoE = 0;
212                         i++;

```

```

211     }
212     } else {
213         resultado.add(new EstabelecimentoResumido(
214             tds.get(0).text(),
215             tds.get(1).text(),
216             municipio.getCodigo(),
217             URL PRINCIPAL + tds.get(0).getElementsByTag("a").attr("href")));
218     }
219     }
220     }
221     }
222     } catch (IOException ex) {
223         qtdeTentativasRequisicao++;
224         System.err.println("-----");
225         System.err.println("Status Code : " + statusCode);
226         System.err.println("Erro - tentativa com erro " + qtdeTentativasRequisicao + " - Tempo limite de
requisição atingido.");
227         System.err.println("Não foi possível conectar a página de estabelecimentos do município de: " +
municipio.getNome());
228         if (qtdeTentativasRequisicao > 4) {
229             qtdeTentativasRequisicao = 0;
230             timeout += 100;
231             System.err.println("Atenção! Verifique sua conexão com a internet");
232             System.err.println("Time out aumentado para " + timeout);
233             System.err.println("Aguardando 5s até a próxima tentativa.");
234             try
235             {
236                 Thread.sleep(5000);
237             } catch (InterruptedException ex1) {
238                 System.err.println("Atenção! Algo deu errado e não será possível aguardando 5s até a próxima
tentativa.");
239             }

```

```
239     }
240     }
241     }
242     return resultado;
243 }
244
245 /**
246  * Método que retorna uma Collection do tipo ArrayList de objetos do tipo
247  * Estabelecimento que contém informações detalhadas do estabelecimento
248  *
249  * @param estabelecimentos ArrayList de objetos do tipo
250  * EstabelecimentoResumido
251  * @return ArrayList de objetos do tipo Estabelecimento
252  */
253 public ArrayList<Estabelecimento> getEstabelecimentosDoMunicipioDetalhados(ArrayList<EstabelecimentoResumido>
estabelecimentos) {
254     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
255     int qtdeTentativasRequisicao = 0;
256     int timeout = 1000;
257     for (int i = 0; i < estabelecimentos.size(); i++) {
258         int statusCode = -1;
259         Document paginaHtmlEstabelecimentoDetalhado = null;
260         Response respostaServidor = null;
261         try {
262             respostaServidor = (Response) Jsoup
263                 .connect(estabelecimentos.get(i).getLinkDaPagina())
264                 .ignoreContentType(true)
265                 .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
266                 .referrer("http://www.google.com")
267                 .timeout(timeout)
268                 .followRedirects(true)
```



```

269         .execute();
270
271         statusCode = respostaServidor.getStatusCode();
272         System.out.println("Iteração: " + i);
273         System.out.println("Estabelecimento: " + estabelecimentos.get(i).getNome());
274         System.out.println("Estabelecimento: " + estabelecimentos.get(i).getLinkDaPagina());
275         System.out.println("StatusCode : " + statusCode);
276         if (statusCode == 200) {
277             timeout = 500; qtdeTentativasRequisicao = 0;
278             paginaHtmlEstabelecimentoDetalhado = respostaServidor.parse();
279             Elements tables = paginaHtmlEstabelecimentoDetalhado.getElementsByTag("table");
280             System.out.println("Tamanho da tabela: " + tables.size());
281             Elements trs = null;
282             String tipoEstab = null;
283             boolean incluirEstabelecimento = true;
284
285             if (tables.size() < 7) {
286                 int iVelho = i;
287                 if (i == iVelho) {
288                     qtdeTentativasRequisicao++;
289                 }
290                 System.err.println("-----");
291                 trs = tables.get(0).getElementsByTag("tr");
292                 System.err.println(trs.get(1).text());
293                 Svstem.err.println("Erro iteração " + i + " tentativa " + qtdeTentativasRequisicao + " - A
294                 página
295                 está fora do ar.");
296                 System.err.println("A página do estabelecimento " + estabelecimentos.get(i).getNome() + "
297                 System.err.println("-----");
298                 if (qtdeTentativasRequisicao > 5) {
299                     System.err.println("Atenção! Aguardando 5s até a próxima tentativa.");
300                     try {

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
298 Thread.sleep(5000);
299     }
300     catch (InterruptedException ex1) {
301     }
302     }
303     i--;
304     } else {
305     if (tables.size() > 7) {
306     trs = tables.get(5).getElementsByTag("tr");
307     tipoEstab = trs.get(9).getElementsByTag("td").get(0).text(); incluirEstabelecimento
308     = isEstabelecimentoFisico(tipoEstab); System.out.println("Tipo estabelecimento: " +
309     tipoEstab); System.out.println(isEstabelecimentoFisico(tipoEstab) ?
310     "Estabelecimento válido" :
311     "Estabelecimento inválido");
312     if (incluirEstabelecimento == false) {
313     System.out.println("-----");
314     }
315     } else {
316     trs = tables.get(4).getElementsByTag("tr");
317     tipoEstab = trs.get(9).getElementsByTag("td").get(0).text(); incluirEstabelecimento
318     = isEstabelecimentoFisico(tipoEstab); System.out.println("Tipo estabelecimento: " +
319     tipoEstab); System.out.println(isEstabelecimentoFisico(tipoEstab) ?
320     "Estabelecimento válido" :
321     "Estabelecimento inválido");
322     if (incluirEstabelecimento == false) {
323     System.out.println("-----");
324     }
325     }
326     if (incluirEstabelecimento) {
327     estabelecimentosValidos++;
328     Estabelecimento estabelecimento = new Estabelecimento();
329     }
```

```

327 estabelecimento.setCnes(Integer.parseInt(estabelecimentos.get(i).getCnes()));
328 String nomeFantasia = trs.get(1).getElementsByTag("td").get(0).text();
329 estabelecimento.setNomeFantasia(nomeFantasia.length() == 0 ? "_" : nomeFantasia);
330 String razaoSocial = trs.get(3).getElementsByTag("td").get(0).text();
331 estabelecimento.setRazaoSocial(razaoSocial.length() == 0 ? "_" : razaoSocial);
332 nomeLogradouro = trs.get(5).getElementsByTag("td").get(0).text();
333 estabelecimento.setLogradouro(nomeLogradouro.length() == 0 ? "_" : nomeLogradouro);
334 String numero = trs.get(5).getElementsByTag("td").get(1).text();
335 estabelecimento.setNumero(numero.length() == 0 ? "S/N" : numero);
336 String telefone = trs.get(5).getElementsByTag("td").get(2).text();
337 estabelecimento.setTelefone(telefone.length() == 0 ? "_" : telefone);
338 String complemento = trs.get(7).getElementsByTag("td").get(0).text();
339 estabelecimento.setComplemento(complemento.length() == 0 ? "_" : complemento);
340 String bairro = trs.get(7).getElementsByTag("td").get(1).text();
341 estabelecimento.setBairro(bairro.length() == 0 ? "_" : bairro);
342 String cep = trs.get(7).getElementsByTag("td").get(2).text();
343 estabelecimento.setCep(cep.length() == 0 ? "_" : cep);
344 String municipio = trs.get(7).getElementsByTag("td").get(3).text();
345 limparNomeMunicipio(municipio); estabelecimento.setMunicipio(municipio.length() ==
346 0 ? "_" : municipio);
347 estabelecimento.setCodigoMunicipio(estabelecimentos.get(i).getCodigoMunicipio());
348 String estado = trs.get(7).getElementsByTag("td").get(4).text();
349 estabelecimento.setEstado(estado.length() == 0 ? "_" : estado);
350 String tipoEstabelecimento = trs.get(9).getElementsByTag("td").get(0).text();
351
352 tipoEstabelecimento);
353 System.out.println("Detalamentos_____");
354 System.out.println("Cnes: " + estabelecimento.getCnes());
355 System.out.println("Razão
social: " + estabelecimento.getRazaoSocial());
356 System.out.println("Nome fantasia: " +
estabelecimento.getNomeFantasia());
357 System.out.println("Tipo estabelecimento: " +
estabelecimento.getTipoEstabelecimento());
358

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
357 System.out.println("Logradouro: " + estabelecimento.getLogradouro());
358 System.out.println("Numero: " + estabelecimento.getNumero());
359 System.out.println("Bairro: " + estabelecimento.getBairro());
360 System.out.println("Complemento: " + estabelecimento.getComplemento());
361 System.out.println("CEP: " + estabelecimento.getCep());
362 System.out.println("Cidade: " + estabelecimento.getMunicipio());
363 System.out.println("Telefone: " + estabelecimento.getTelefone());
364 System.out.println("Estado: " + estabelecimento.getEstado());
365
366     if (estabelecimentos.get(i).getLinkDaPagina().toString().length() == 117) {
367
368         estabelecimento.setProfissionaisVinculados(getProfissionaisDoEstabelecimento(estabelecimentos.get(i).getIdConsulta()));
369
370         estabelecimento.setServicosOferecidos(getServicosDoEstabelecimento(estabelecimentos.get(i).getIdConsulta()));
371
372         estabelecimento.setAtendimentosPrestados(getAtendimentosDoEstabelecimento(estabelecimentos.get(i).getIdConsulta()));
373     } else {
374         System.err.println("Estabelecimento desativado e com link bloqueado");
375     }
376
377     Double[] coordenadas = getGeocod(nomeLogradouro, numero, bairro, cep, municipio, estado);
378     estabelecimento.setLatitude(coordenadas[0]);
379     estabelecimento.setLongitude(coordenadas[1]);
380     resultado.add(estabelecimento);
381     System.out.println("-----");
382 }
383
384 } catch (IOException ex) {
385     int iVelho = i;
386     if (i == iVelho) {
387         qtdeTentativasRequisicao++;
388     }
389 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
385     }
386     System.err.println("-----");
387     System.err.println("StatusCode : " + statusCode);
388     System.err.println("Erro ao conectar a página de informações sobre o estabelecimento");
389     System.err.println("Erro iteração " + i + " tentativa com erro " + qtdeTentativasRequisicao + " -
Tempo limite de requisição atingido.");
390     System.err.println("-----");
391     if (qtdeTentativasRequisicao > 5) {
392         timeout += 100;
393         System.err.println("Timeout aumentado para: " + timeout);
394         System.err.println("Atenção! Aguardando 5s até a próxima tentativa.");
395         try {
396             Thread.sleep(5000);
397         } catch (InterruptedException ex1) {
398             System.err.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
tentativa.");
399         }
400     }
401     i--;
402 }
403 }
404 System.err.println("Estabelecimentos válidos: " + estabelecimentosValidos);
405 System.err.println("Coordenadas válidas OSM: " + coordLocalizadasOSM);
406 System.err.println("Coordenadas válidas GM: " + coordLocalizadasGM);
407 System.err.println("Estabelecimentos sem coordenadas geográficas: " + (estabelecimentosValidos -
(coordLocalizadasGM + coordLocalizadasOSM)));
408
409 estabelecimentosValidos = 0;
410 coordLocalizadasOSM = 0;
411 coordLocalizadasGM = 0;
412 return resultado;
```

```
413     }
414
415     /**
416      * Método interno que retorna um booleano que identifica se um
417      * estabelecimento de saúde é físico, fixo e presta algum tipo de
418      * atendimento pessoal e local
419      *
420      * @param String contendo o tipo de estabelecimento a ser analisado pelo
421      * método
422      * @return boolean que representa se o estabelecimento atende os parametros
423      * pre-definidos
424      */
425     private boolean isEstabelecimentoFisico(String tipo) {
426         boolean fim = false;
427         boolean resultado = false;
428         int i = 0;
429         while (!fim && i < estabelecimentosFisicos.size()) {
430             if (estabelecimentosFisicos.get(i).equalsIgnoreCase(tipo)) {
431                 resultado = true;
432                 fim = true;
433             }
434             i++;
435         }
436         return resultado;
437     }
438
439     private String isProfissionalValido(String cbo) {
440         String resultado = "";
441         if (cbo.length() < 7) {
442             System.out.println(cbo);
443         } else {
444             ...
445         }
446     }
447 }
```

```

444     String familiaCB0 = cbo.trim().substring(0, 6);
445     if (familiaCB0.startsWith("2212"))
446         || familiaCB0.startsWith("2231")
447         || familiaCB0.startsWith("2232")
448         || familiaCB0.startsWith("2236")
449         || familiaCB0.startsWith("2237")
450         || familiaCB0.startsWith("2238")
451         || familiaCB0.startsWith("2239")
452         || familiaCB0.startsWith("2241")
453         || familiaCB0.startsWith("2251")
454         || familiaCB0.startsWith("2252")
455         || familiaCB0.startsWith("2253")
456         || familiaCB0.startsWith("2261")
457         || familiaCB0.startsWith("2263")
458         || familiaCB0.startsWith("2515")) {
459         resultado = familiaCB0;
460     }
461 }
462 return resultado;
463 }
464
465 private String limparNomeMunicipio(String nomeSujo) {
466     String nomeLimp0 = null;
467     int i = 0;
468     int idApostrofo = 0;
469     boolean ok = false;
470     while (!ok) {
471         if (nomeSujo.charAt(i) == ' ') {
472             ok = true;
473             nomeLimp0 = nomeSujo.substring(0, i - 1);
474         }
475     }

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
475         i++;
476     }
477     idApostrofo = nomeLimpo.indexOf("'");
478     if (idApostrofo > 0) {
479         nomeLimpo = nomeLimpo.replace("'", " ");
480     }
481     return nomeLimpo;
482 }
483
484 private ArrayList<Profissional> getProfissionaisDoEstabelecimento(String idConsulta) {
485     int qtdeTentativasRequisicao = 1;
486     int qtdeTentativasRequisicaoP = 1;
487     int statusCode = -1;
488     boolean fim = false;
489     int timeOut = 1000;
490     ArrayList<Profissional> resultado = new ArrayList<Profissional>();
491     Document paginaHtmlProfissionais = null;
492     Response respostaServidor = null;
493     System.out.println("Conectando a página dos profissionais++++++++++++++++++++++++++++++++++++++++");
494     while (fim == false) {
495         try {
496             respostaServidor = (Response) Jsoup
497                 .connect(URL_PRINCIPAL + FUNCAO_PROFFISIONAIS_ESTABELECIMENTO + idConsulta)
498                 .ignoreContentType(true)
499                 .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
500                 .referrer("http://www.google.com")
501                 .timeout(timeOut)
502                 .followRedirects(true)
503                 .execute();
504             statusCode = respostaServidor.statusCode();
505             System.out.println("StatusCode : " + statusCode);
506         }
507     }
508 }
```


E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
506     if (statusCode == 200) {
507         timeOut = 500; qtdeTentativasRequisicao = 1;
508         paginaHtmlProfissionais = respostaServidor.parseO();
509         Elements trs = paginaHtmlProfissionais.select(".gradeA");
510
511         if (trs.size() == 0) {
512             System.out.println("Não há profissionais cadastrados");
513             fim = true;
514         } else {
515             Elements tds = trs.select("td");
516             System.out.println("Tamanho da tabela de profissionais: " + tds.size());
517             if (paginaHtmlProfissionais.select("table").size() < 6) {
518                 fim = false;
519                 System.err.println("Erro - Site fora do ar. Uma nova tentativa será realizada");
520             } else {
521                 for (int i = 0; i < trs.size(); i++) {
522                     qtdeTentativasRequisicaoP++;
523                     if (trs.get(i).select("td").size() < 6) {
524                         System.err.println("Erro - QTDE de TDS: " + trs.get(i).select("td").size());
525                         System.err.println(trs.get(i).select("td"));
526                         System.err.println("Erro - Problemas ao carregar profissional - linha: " + i);
527                         System.err.println("Erro - Tentativa " + qtdeTentativasRequisicaoP);
528                         i--;
529                         if (qtdeTentativasRequisicaoP > 5) {
530                             System.err.println("Impossível ler a linha: " + i + ", ela será descartada.");
531                             qtdeTentativasRequisicaoP = 0;
532                             i++;
533                         }
534                     } else {
535                         String r1 = trs.get(i).select("td").get(5).text();
536                         if (r1.length() > 9) {
537                             --

```

```

537 String cbo = isProfissionalValido(r1);
538 String nome = trs.get(i).select("td").get(0).text();
539 String descCB0 = r1.substring(9);
540 if (cbo.length() > 0) {
541     System.out.println("Linha: " + i + " Profissional: " + cbo + " | " +
nome);
542     Profissional p = new Profissional();
543     p.setCbo(cbo);
544     p.setNome(nome);
545     p.setCboDescricao(descCB0);
546     resultado.add(p);
547 }
548 } else {
549     System.err.println("Erro de leitura na linha: " + i + "ela será descartada");
550     System.out.println(r1);
551 }
552 }
553 }
554 fim = true;
555 }
556 }
557 } catch (IOException ex) {
558     qtdeTentativasRequisicao++;
559     System.err.println("-----");
560     System.err.println("Status Code : " + statusCode);
561     System.err.println("Erro ao conectar a página de profissionais do estabelecimento");
562     System.err.println("Tentativa " + qtdeTentativasRequisicao + " - Tempo limite de requisição
atingido.");
563     System.err.println("-----");
564     if (qtdeTentativasRequisicao > 5) {
565         --

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
566         timeOut += 100;
567         System.err.println("Atenção! Aguardando 5s até a próxima tentativa.");
568         System.err.println("Timeout aumentado para: " + timeOut);
569         qtdeTentativasRequisicao = 0;
570         try
571         {
572             Thread.sleep(5000);
573         } catch (InterruptedException ex1) {
574             System.err.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
575             tentativa.");
576         }
577         }
578         return resultado;
579     }
580
581     private ArrayList<ServicoClassificacao> getServicosDoEstabelecimento(String idConsulta) {
582         int qtdeTentativasRequisicao = 1;
583         int qtdeTentativasRequisicaoS = 1;
584         int statusCode = -1;
585         boolean fim = false;
586         int timeOut = 1000;
587         ArrayList<ServicoClassificacao> resultado = new ArrayList<ServicoClassificacao>();
588         Document paginaHtmlServicos = null;
589         Response respostaServidor = null;
590         System.out.println("Conectando a página Serviços XXXXXXXXXXXXXXXXXXXXXXXXXX");
591         String servClassOld = "";
592         while (fim == false) {
593             try {
594                 respostaServidor = (Response) Jsoup
595                     .connect(URL_PRINCIPAL + FUNCAO_SERVICO_CLASSIFICACAO + idConsulta)
596             }
597         }
598     }
599 }
```

```
596 .ignoreContentType(true)
597 .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
598 .referrer("http://www.google.com")
599 .timeOut(timeOut)
600 .followRedirects(true)
601 .execute();
602
603 statusCode = respostaServidor.statusCode();
604 System.out.println("Status Code : " + statusCode);
605 if (statusCode == 200) {
606     timeOut = 500; qtdeTentativasRequisicao = 1;
607     paginaHtmlServicos = respostaServidor.parse();
608     Elements tables = paginaHtmlServicos.select("table");
609
610     if (tables.size() == 1) {
611         fim = false;
612         System.err.println("Erro - Site fora do ar. Uma nova tentativa será realizada");
613     } else {
614         Elements trs = tables.get(tables.size() - 4).select("tr");
615         Elements tds = trs.select("td");
616         if (trs.size() <= 0) {
617             System.out.println("Não possui serviços cadastrados.");
618             fim = true;
619         } else {
620             System.out.println("Tamanho da tabela de serviços: " + tds.size());
621             for (int i = 0; i < trs.size(); i++) {
622                 qtdeTentativasRequisicaoS++;
623                 if (trs.get(i).select("td").size() < 5) {
624                     System.err.println("Erro - QTDE de TDS: " + trs.get(i).select("td").size());
625                     System.err.println(trs.get(i).select("td"));
626                     System.err.println("Erro - Problemas ao carregar serviços - linha: " + i);
627                     System.err.println("Erro - Tentativa " + qtdeTentativasRequisicaoS);
628                 }
629             }
630         }
631     }
632 }
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```

627     i--;
628     if (qtdeTentativasRequisicaoS > 5) {
629         System.err.println("Impossível ler a linha: " + i + ", ela será descartada.");
630         qtdeTentativasRequisicaoS = 0;
631         i++;
632     }
633     } else {
634         String servClas = trs.get(i).select("td").get(0).text();
635         if (servClassOld.equalsIgnoreCase(servClas)) {
636             System.out.println("Serviço e Classificação em duplicidade: " + servClas + "
e será descartado");
637         } else {
638             ServicoClassificacao s = new ServicoClassificacao();
639             String descServ = trs.get(i).select("td").get(1).text();
640             String descClass = trs.get(i).select("td").get(2).text();
641             String serv = servClas.substring(0, 3);
642             String classif = servClas.substring(6, 9);
643             System.out.println("Serv: " + serv);
644             System.out.println("Class: " + classif);
645             System.out.println("Desc serv: " + descServ);
646             System.out.println("Desc class: " + descClass);
647             s.setServico(Integer.parseInt(serv));
648             s.setClassificacao(Integer.parseInt(classif));
649             resultado.add(s);
650         }
651         servClassOld = servClas;
652     }
653     }
654     fim = true;
655 }
656 ---

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
657     }
658     } catch (IOException ex) {
659         qtdeTentativasRequisicao++;
660         System.err.println("-----");
661         System.err.println("Status Code : " + statusCode);
662         System.err.println("Erro ao conectar a página de Serviços do estabelecimento");
663         System.err.println("Tentativa " + qtdeTentativasRequisicao + " - Tempo limite de requisição
atingido.");
664         System.err.println("-----");
665         if (qtdeTentativasRequisicao > 5) {
666             timeout += 100;
667             System.err.println("Atenção! Aguardando 5s até a próxima tentativa.");
668             System.err.println("Timeout aumentado para: " + timeout);
669             qtdeTentativasRequisicao = 0;
670             try
671             {
672                 Thread.sleep(5000);
673                 } catch (InterruptedException ex1) {
674                     System.err.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
tentativa.");
675                 }
676             }
677         }
678         return resultado;
679     }
680
681     private ArrayList<AtendimentoPrestado> getAtendimentosDoEstabelecimento(String idConsulta) {
682         int qtdeTentativasRequisicao = 1;
683         int qtdeTentativasRequisicaoA = 1;
684         int statusCode = -1;
685         boolean fim = false;
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
686 int timeOut = 1000;
687 ArrayList<AtendimentoPrestado> resultado = new ArrayList<AtendimentoPrestado>();
688 Document paginaAtendimentosPrestados = null;
689 Response respostaServidor = null;
690 System.out.println("Conectando a página de Atendimentos prestados %%%%%%%%%%%");
691 while (fim == false) {
692     try {
693         respostaServidor = (Response) Jsoup
694             .connect(URL_PRINCIPAL + FUNCAO_ATENDIMENTOS_PRESTADOS + idConsulta)
695             .ignoreContentType(true)
696             .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
697             .referrer("http://www.google.com")
698             .timeout(timeOut)
699             .followRedirects(true)
700             .execute();
701         statusCode = respostaServidor.statusCode();
702         System.out.println("StatusCode : " + statusCode);
703         System.out.println("Link da consulta: " + URL_PRINCIPAL + FUNCAO_ATENDIMENTOS_PRESTADOS +
idConsulta);
704         if (statusCode == 200) {
705             timeOut = 500;
706             paginaAtendimentosPrestados = respostaServidor.parse();
707             Elements tabelas = paginaAtendimentosPrestados.select("table");
708             System.err.println("Qtde de tabelas: " + tabelas.size());
709             if (tabelas.size() == 1) {
710                 fim = false;
711                 System.err.println("Erro - Site fora do ar. Uma nova tentativa será realizada");
712             } else if (tabelas.size() == 3) {
713                 System.out.println("Não possui Tipos de atendimentos cadastrados.");
714                 fim = true;
715             } else {
716                 --

```

```

716     qtdeTentativasRequisicao = 1;
717     Elements trs = tabelas.get(5).select("tr");
718     for (int i = 0; i < trs.size(); i++) {
719         qtdeTentativasRequisicaoA++;
720         Elements tds = trs.get(i).select("td");
721         if (tds.size() < 2) {
722             System.err.println("Erro - QTDE de TDS: " + trs.get(i).select("td").size());
723             System.err.println(trs.get(i).select("td"));
724             System.err.println("Erro - Problemas ao carrega atendimento - linha: " + i);
725             System.err.println("Erro - Tentativa " + qtdeTentativasRequisicaoA);
726             i--;
727             if (qtdeTentativasRequisicaoA > 5) {
728                 System.err.println("Impossível ler a linha: " + i + ", ela será descartada.");
729                 qtdeTentativasRequisicaoA = 0;
730                 i++;
731             }
732             } else {
733                 AtendimentoPrestado a = new AtendimentoPrestado();
734                 String tipoAtdm = tds.get(0).text();
735                 String conven = tds.get(1).text();
736                 a.setTipoAtendimento(tipoAtdm);
737                 a.setConvenio(conven);
738                 resultado.add(a);
739                 System.out.println("Tipo de atendimento: " + tipoAtdm + " | Convênio: " + conven);
740             }
741         }
742         fim = true;
743     }
744     } catch (IOException ex) {
745         qtdeTentativasRequisicao++;
746     }

```


E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
747 System.err.println("-----");
748 System.err.println("StatusCode : " + statusCode);
749 System.err.println("Erro ao conectar a página de Atendimentos prestados do estabelecimento");
750 System.err.println("Tentativa " + qtdeTentativasRequisicao + " - Tempo limite de requisição
atingido.");
751 System.err.println("-----");
752 if (qtdeTentativasRequisicao > 5) {
753     timeOut += 100;
754     System.err.println("Atenção! Aguardando 5s até a próxima tentativa.");
755     System.err.println("Timeout aumentado para: " + timeOut);
756     qtdeTentativasRequisicao = 0;
757     try {
758         Thread.sleep(5000);
759     } catch (InterruptedException ex1) {
760         System.err.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
tentativa.");
761     }
762 }
763 }
764 }
765 return resultado;
766 }
767
768 private Double[] getGeocodGoogleMaps(String endereco) {
769     int qtdeTentativasRequisicao = 1;
770     int statusCode = -1;
771     boolean fim = false;
772     int timeOut = 1000;
773     Double[] resultado = new Double[2];
774     Document xmlDoc = null;
775     Connection.Response respostaServidor = null;
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
776 endereco = endereco.replace(" ", "+");
777 System.out.println(endereco);
778 while (fim == false) {
779     try {
780         respostaServidor = (Connection.Response) Jsoup
781             .connect("https://maps.googleapis.com/maps/api/geocode/xml?address=" + endereco + "&key=" +
googleApiKey)
782             .ignoreContentType(true)
783             .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
784             .referrer("http://www.google.com")
785             .timeout(timeout)
786             .followRedirects(true)
787             .execute();
788         statusCode = respostaServidor.statusCode();
789         System.out.println("StatusCode : " + statusCode);
790         if (statusCode == 200) {
791             timeout = 500;
792             fim = true;
793             xmlDoc = respostaServidor.parse();
794             Elements latLon = xmlDoc.select("location");
795             System.out.println("Quantidade de resultados: " + latLon.size());
796             if (latLon.size() > 0) {
797                 if (latLon.size() > 1) {
798                     resultado[0] = Double.parseDouble(latLon.get(0).select("lat").text());
799                     resultado[1] = Double.parseDouble(latLon.get(0).select("lng").text());
800                 } else {
801                     resultado[0] = Double.parseDouble(latLon.select("lat").text());
802                     resultado[1] = Double.parseDouble(latLon.select("lng").text());
803                 }
804                 System.out.println("GeocodGoogleMaps " + resultado[0] + " " + resultado[1]);
805             }
806         }
807     }
808 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
806     }
807     } catch (IOException ex) {
808         qtdeTentativasRequisicao++;
809         System.err.println("-----");
810         System.err.println("StatusCode : " + statusCode);
811         System.err.println("Erro ao conectar a página de Geocoding do Google Mapas");
812         System.err.println("Tentativa " + qtdeTentativasRequisicao + " - Tempo limite de requisição
atingido.");
813         System.err.println("-----");
814         if (qtdeTentativasRequisicao > 5) {
815             timeout += 100;
816             System.err.println("Atenção! Aguardando 5s até a próxima tentativa.");
817             System.err.println("Timeout aumentado para: " + timeout);
818             qtdeTentativasRequisicao = 0;
819             try
820             {
821                 Thread.sleep(5000);
822             } catch (InterruptedException ex1) {
823                 System.err.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
tentativa.");
824             }
825         }
826     }
827     }
828     return resultado;
829 }
830
831 private Double[] getGeocodOpenStreetMaps(String endereco) {
832     int qtdeTentativasRequisicao = 1;
833     int statusCode = -1;
834     boolean fim = false;
```

```

835 int timeOut = 1000;
836 Double[] resultado = new Double[2];
837 Document xmlDoc = null;
838 Connection.Response respostaServidor = null;
839 endereco = endereco.replace(" ", "%20");
840 System.out.println(endereco);
841 while (fim == false) {
842     try {
843         respostaServidor = (Connection.Response) Jsoup
844             .connect("http://nominatim.openstreetmap.org/search/" + endereco.toLowerCase() +
"?format=xml")
845             .ignoreContentType(true)
846             .userAgent("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1")
847             .referrer("http://www.google.com")
848             .timeout(timeOut)
849             .followRedirects(true)
850             .execute();
851         statusCode = respostaServidor.statusCode();
852         System.out.println("Status Code : " + statusCode);
853         if (statusCode == 200) {
854             timeOut = 1000;
855             fim = true;
856             xmlDoc = respostaServidor.parse(); Elements
857             latLon = xmlDoc.select("place");
858             System.out.println("Quantidade de resultados: " + latLon.size());
859             if (latLon.size() > 0) {
860                 if (latLon.size() > 1) {
861                     resultado[0] = Double.parseDouble(latLon.get(0).attr("lat"));
862                     resultado[1] = Double.parseDouble(latLon.get(0).attr("lon"));
863                 } else {
864                     resultado[0] = Double.parseDouble(latLon.attr("lat"));

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ConversorHtmlParaJava.java

```
865         resultado[1] = Double.parseDouble(latLon.attr("lon"));
866     }
867     System.out.println("GeocodOpenStreetMaps " + resultado[0] + " " + resultado[1]);
868 }
869 } catch (IOException ex) {
870     qtdeTentativasRequisicao++;
871     System.err.println("-----");
872     System.err.println("StatusCode : " + statusCode);
873     System.err.println("Erro ao conectar a página de Geocoding do Open Street Maps");
874     System.err.println("Tentativa " + qtdeTentativasRequisicao + " - Tempo limite de requisição
atingido.");
875     System.err.println("-----");
876     if (qtdeTentativasRequisicao > 5) {
877         timeout = timeout + 1000;
878         System.err.println("Atenção! Aguardando 5s até a próxima tentativa.");
879         System.err.println("Timeout aumentado para: " + timeout);
880         qtdeTentativasRequisicao = 0;
881         try
882         {
883             Thread.sleep(5000);
884         } catch (InterruptedException ex1) {
885             System.err.println("Atenção! Algo deu errado e não será possível Aguardando 5s até a próxima
tentativa.");
886         }
887     }
888 }
889 }
890     return resultado;
891 }
892
893     private String[] extrairTipoLogradouro(String logradouro) {
```

```

894 String[] resultado = new String[2];
895 boolean ok = false;
896 int i = 0;
897 while (!ok && i < logradouro.length()) {
898     if (logradouro.charAt(i) == ' ') {
899         resultado[0] = logradouro.substring(0, i); //tipo de logradouro
900         resultado[1] = logradouro.substring(i + 1, logradouro.length()); //nome do logradouro
901         ok = true;
902     }
903     i++;
904 }
905 if (resultado[0] == null) {
906     resultado[0] = "";
907     resultado[1] = logradouro.trim();
908 } else {
909     resultado[0].trim();
910     resultado[1].trim();
911 }
912 return resultado;
913 }
914
915 private Double[] getGeocod(String logradouro, String numero, String bairro, String cep, String municipio, String
estado) {
916     Double[] resultado = new Double[2];
917     String[] logradouroFormatado = extrairTipoLogradouro(logradouro);
918     String enderecoOSM = numero + " " + logradouroFormatado[1] + " " + logradouroFormatado[0] + ", " + municipio
+ ", " + estado + ", " + cep;
919     String enderecoGM = logradouro + " " + numero + " " + bairro + " " + cep + " " + municipio + " " + estado;
920     resultado = getGeocodOpenStreetMaps(enderecoOSM);
921     if (resultado[0] != null) {
922         System.out.println("Origem das coordenadas: OpenStreetMaps");

```

```
923     coordLocalizadasOSM++;
924     } else {
925         resultado = getGeocodGoogleMaps(enderecoGM);
926         if (resultado[0] != null) {
927             System.out.println("Origem das coordenadas: GoogleMaps");
928             coordLocalizadasGM++;
929         } else {
930             System.out.println("Origem das coordenadas: Coordenads não localizadas");
931         }
932     }
933     return resultado;
934 }
935
936 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar dimon/jwebcrawlercnes/principal/EstabelecimentoDAO.java

```

1 package br.com.jucemar_dimon.jwebcrawlercnes.principal;
2
3 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.Estabelecimento;
4 import java.sql.Connection;
5 import java.sql.DriverManager;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.sql.Statement;
9 import java.util.ArrayList;
10 import java.util.logging.Level;
11 import java.util.logging.Logger;
12
13 /**
14  * @author Jucemar
15  */
16 public class EstabelecimentoDAO {
17
18     private Connection c = null;
19     private Statement stmt = null;
20
21     public EstabelecimentoDAO() {
22         try {
23             Class.forName("org.sqlite.JDBC");
24             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
25             System.out.println("Conexão com o banco de dados estabelecida com sucesso");
26         } try {
27             String tabelaEstabelecimento = "CREATE TABLE IF NOT EXISTS estabelecimento "
28                 + "(cnes INTEGER NOT NULL,"
29                 + " razao_social TEXT,"
30                 + " nome_fantasia TEXT,"
31                 + " telefone TEXT,"

```



```

32 + " nome_logradouro TEXT NOT NULL,"
33 + " numero TEXT,"
34 + " complemento TEXT,"
35 + " bairro TEXT,"
36 + " municipio TEXT NOT NULL,"
37 + " codigo_municipio TEXT NOT NULL,"
38 + " estado TEXT NOT NULL,"
39 + " cep TEXT NOT NULL,"
40 + " latitude REAL,"
41 + " longitude REAL,"
42 + " tipo_estabelecimento TEXT NOT NULL,"
43 + " PRIMARY KEY(cnes)):";
44 String index = "CREATE INDEX IF NOT EXISTS estabelecimento_PKIndex ON estabelecimento (cnes)";
45 stmt = c.createStatement();
46 stmt.executeUpdate(tabelaEstabelecimento);
47 stmt.executeUpdate(index);
48 stmt.close();
49 c.close();
50 } catch (Exception e) {
51     System.err.println(e.getClass().getName() + ": " + e.getMessage());
52 }
53 System.out.println("Tabela de estabelecimentos criada com sucesso");
54 } catch (ClassNotFoundException ex) {
55     Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
56 } catch (SQLException ex) {
57     Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
58 }
59 }
60 public boolean inserirEstabelecimento(Estabelecimento estabelecimento) {
61     boolean novoEstabelecimento = isEstabelecimentoExiste(estabelecimento.getCnes());
62     --

```

```

63     if (novoEstabelecimento) {
64         try {
65             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
66             this.stmt = this.c.createStatement();
67         } try {
68             String sql = "INSERT INTO estabelecimento("
69                 + "cnes, razao_social, nome_fantasia, nome_logradouro, numero, complemento, bairro,
municipio, codigo_municipio, estado, telefone, cep, latitude, longitude, tipo_estabelecimento)"
70                 + "VALUES("
71                 + estabelecimento.getCnes() + "°, "°"
72                 + estabelecimento.getRazaoSocial().trim() + "°, "°"
73                 + estabelecimento.getNomeFantasia().trim() + "°, "°"
74                 + estabelecimento.getLogradouro().trim() + "°, "°"
75                 + estabelecimento.getNumero().trim() + "°, "°"
76                 + estabelecimento.getComplemento().trim() + "°, "°"
77                 + estabelecimento.getBairro().trim() + "°, "°"
78                 + estabelecimento.getMunicipio().trim() + "°, "°"
79                 + estabelecimento.getCodigoMunicipio().trim() + "°, "°"
80                 + estabelecimento.getEstado().trim() + "°, "°"
81                 + estabelecimento.getTelefone().trim() + "°, "°"
82                 + estabelecimento.getCep().trim() + "°, "°"
83                 + estabelecimento.getLatitude() + "°, "°"
84                 + estabelecimento.getLongitude() + "°, "°"
85                 + estabelecimento.getTipoEstabelecimento().trim()
86                 + "°"); "°";
87             System.out.println(sql);
88             this.stmt.executeUpdate(sql);
89
90         } catch (SQLException e) {
91             e.printStackTrace();
92         }

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/EstabelecimentoDAO.java

```
93         stmt.close();
94         c.close();
95     } catch (SQLException ex) {
96         Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
97     }
98 }
99 return novoEstabelecimento;
100 }
101
102 private boolean isEstabelecimentoExiste(int cnes) {
103     boolean novoEstabelecimento = true;
104     ResultSet rs = null;
105     try {
106         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
107         this.stmt = this.c.createStatement();
108         rs = this.stmt.executeQuery("SELECT cnes FROM estabelecimento where cnes=" + cnes + ";");
109         if (rs.next()) {
110             novoEstabelecimento = false;
111         }
112         rs.close();
113         stmt.close();
114         c.close();
115     } catch (SQLException e) {
116     }
117     return novoEstabelecimento;
118 }
119
120 public int atualizarCoordenadasGeograficas(Double latitude, Double longitude, int cnes) {
121     int resultado = -1;
122     try {
123         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
124     }
125 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/EstabelecimentoDAO.java

```
124     this.stmt = this.c.createStatement();
125     resultado = this.stmt.executeUpdate("UPDATE estabelecimento set latitude=" + latitude + ", longitude=" +
longitude + " where cnes=" + cnes + ";");
126     stmt.close();
127     c.close();
128     } catch (SQLException e) {
129         System.err.println("Não foi possível atualizar as coordenadas do estabelecimento: " + cnes + " " +
e.getMessage());
130     }
131     return resultado;
132 }
133
134     public static final Estabelecimento extrairDoCursor(ResultSet rs) throws SQLException {
135         int cnes = rs.getInt(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.CNES);
136         String razaoSocial =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.RAZAO_SOCIAL);
137         String nomeFantasia =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.NOME_FANTASIA);
138         String telefone =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.TELEFONE);
139         String nomeLogradouro =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.NOME_LOGRADOURO);
140         String numero =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.NUMERO);
141         String complemento =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.COMPLEMENTO);
142         String bairro =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.BAIRRO);
143         String municipio =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.MUNICIPIO);
144         String codigoMunicipio =
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/EstabelecimentoDAO.java

```
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO);
145     String estado =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.ESTADO);
146     String cep = rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.CEP);
147     Double latitude =
rs.getDouble(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.LATITUDE);
148     Double longitude =
rs.getDouble(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.LONGITUDE);
149     String tipoEstabelecimento =
rs.getString(br.com.jucemar_dimon.jwebcrawlercnes.principal.TabelaEstabelecimento.Colunas.TIPO_ESTABELECIMENTO);
150     return new Estabelecimento(cnes, razaoSocial, nomeFantasia, nomeLogradouro, numero, complemento, bairro, cep,
municipio, codigoMunicipio, estado, telefone, tipoEstabelecimento, latitude, longitude);
151 }
152
153 public ArrayList<Estabelecimento> listaTodosEstabelecimentos() {
154     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
155     ResultSet rs = null;
156     try {
157         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
158         this.stmt = this.c.createStatement();
159         rs = this.stmt.executeQuery("SELECT * FROM estabelecimento;");
160         do {
161             Estabelecimento e = extrairDoCursor(rs);
162             resultado.add(e);
163         } while (rs.next());
164         rs.close();
165         stmt.close();
166         c.close();
167     } catch (SQLException e) {
168         System.err.println("Não foi possível realizar a consulta");
169     }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/EstabelecimentoDAO.java

170 return resultado;

171 }

172

173 }

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/JWebCrawlerCNES.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.principal;
2
3 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.AtendimentoPrestado;
4 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.Estabelecimento;
5 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.EstabelecimentoResumido;
6 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.Municipio;
7 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.Profissional;
8 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.ServicoClassificacao;
9 import br.com.jucemar_dimon.jwebcrawlercnes.sigtap.SigTap;
10 import br.com.jucemar_dimon.jwebcrawlercnes.utilitarios.ManipuladorDeArquivos;
11 import java.io.FileNotFoundException;
12 import java.io.IOException;
13 import java.util.ArrayList;
14
15 /**
16  * @author Jucemar
17  */
18 public class JWebCrawlerCNES {
19
20     public static void main(String[] args) throws IOException {
21         ManipuladorDeArquivos bkp = new ManipuladorDeArquivos();
22         ConversorHtmlParaJava conversor = new ConversorHtmlParaJava();
23         try {
24             System.out.println("Iniciando Etapa de captura de dados do Sigtap");
25             SigTap.iniciar();
26         } catch (FileNotFoundException eFNF) {
27             System.out.println(eFNF.getMessage());
28             System.out.println("Um dos arquivos de texto que contém uma das tabelas SIGTAP não foi localizado.");
29             System.out.println("Os arquivos são intitulados de: tb_servico.txt e tb_servico_classificacao.txt");
30             System.out.println("O programa será fechado");
31             System.exit(0);
32         }
33     }
34 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/JWebCrawlerCNES.java

```
32     } catch (IOException eIO) {
33         System.out.println(eIO.getMessage());
34         System.out.println("Houve um problema durante a manipulação dos arquivos de texto que contém as tabelas
SIGTAP");
35         System.out.println("O programa será fechado");
36         System.exit(0);
37     }
38     System.out.println("Iniciando etapa de captura de dados dos Estabelecimentos");
39     ArrayList<Municipio> resultado = bkp.leiaDadosDEArquivoTemporario();
40     if (resultado == null) {
41         resultado = conversor.getMunicipios();
42     }
43     EstabelecimentoDAO eDao = new EstabelecimentoDAO();
44     ServicoClassificacaoEstabelecimentoDAO sDao = new ServicoClassificacaoEstabelecimentoDAO();
45     ProfissionalDAO pDao = new ProfissionalDAO();
46     VinculoProfissionalEstabelecimentoDAO vDao = new VinculoProfissionalEstabelecimentoDAO();
47     AtendimentoPrestadoDAO aDao = new AtendimentoPrestadoDAO();
48     for (Municipio m : resultado) {
49         if (m.getEstabelecimentos() == null) {
50             System.out.println("Os estabelecimentos do município de " + m.getNome() + " ainda não foram salvos");
51             ArrayList<EstabelecimentoResumido> estabResumidos = conversor.getEstabelecimentosDoMunicipio(m);
52             ArrayList<Estabelecimento> estaDetalhados =
conversor.getEstabelecimentosDoMunicipioDetalhados(estabResumidos);
53             m.setEstabelecimentos(estaDetalhados);
54             bkp.salveDadosEmArquivoTemporario(resultado);
55         }
56         System.out.println("Os estabelecimentos do município de " + m.getNome() + " já foram salvos e serão
verificados sua existência na base de dados");
57         bkp.salveDadosEmArquivoTemporario(resultado);
58         for (Estabelecimento e : m.getEstabelecimentos()) {
59             boolean novoEstabelecimento = eDao.inserirEstabelecimento(e);
```


E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/JWebCrawlerCNES.java

```
60         if (novoEstabelecimento) {
61             System.out.println("0 estabelecimento " + e.getNomeFantasia() + " será inserido na base de
dados");
62         for (Profissional p : e.getProfissionaisVinculados()) {
63             int id = pDao.inserirProfissional(p);
64             System.out.println("Id do profissional inserido: " + id);
65             vDao.inserirVinculoDoProfissional(id, e.getCnes());
66         }
67         for (ServicoClassificacao s : e.getServicosOferecidos()) {
68             sDao.inserirServicoClassificacao(e.getCnes(), s);
69         }
70         for (AtendimentoPrestado a : e.getAtendimentosPrestados()) {
71             aDao.inserirAtendimento(e.getCnes(), a);
72         }
73         } else {
74             System.out.println("0 estabelecimento " + e.getNomeFantasia() + " já foi inserido na base de
dados");
75         }
76     }
77 }
78 }
79
80 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ProfissionalDAO.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.principal;
2
3 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.Profissional;
4 import java.sql.Connection;
5 import java.sql.DriverManager;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.sql.Statement;
9 import java.util.logging.Level;
10 import java.util.logging.Logger;
11
12 /**
13  * @author Jucemar
14  */
15 public class ProfissionalDAO {
16
17     private Connection c = null;
18     private Statement stmt = null;
19
20     public ProfissionalDAO() {
21         try {
22             Class.forName("org.sqlite.JDBC");
23             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
24         } catch (Exception e) {
25             String tabelaProfissional = "CREATE TABLE IF NOT EXISTS profissional "
26                 + "(id INTEGER PRIMARY KEY AUTOINCREMENT,"
27                 + " nome TEXT NOT NULL,"
28                 + " cbo TEXT NOT NULL,"
29                 + " cbo_descricao TEXT NOT NULL);";
30             String index = "CREATE INDEX IF NOT EXISTS profissional_PKIndex ON profissional (id);";
31             stmt = c.createStatement();
32         }
33     }
34 }
```


E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar dimon/jwebcrawlercnes/principal/ProfissionalDAO.java

```
62         + profissional.getCboDescricaoO.trimO + "");";
63     System.out.println(sql);
64     this.stmt.executeUpdate(sql);
65     rs = this.stmt.executeQuery(lastId);
66     ultimoId = rs.getInt(1);
67     rs.close();
68 }
69 stmt.close();
70 c.close();
71     catch      (SQLException      ex)      {
72         Logger.getLogger(ProfissionalDAO.class.getName()).log(Level.SEVERE, null, ex);
73     }
74     return ultimoId;
75 }
76
77 private int isProfissionalExiste(Profissional profissional) {
78     int idprofissional = 0;
79     ResultSet rs = null;
80     try {
81         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
82         this.stmt = this.c.createStatement();
83         String buscaProfissional = "SELECT id FROM profissional WHERE nome=" + profissional.getNomeO + "" and
cbo=" + profissional.getCboO + """;
84         System.out.println(buscaProfissional);
85         rs = this.stmt.executeQuery(buscaProfissional);
86         if (rs.next()) {
87             idprofissional = rs.getInt("id");
88             System.out.println("Profissional encontrado na base de dados: " + profissional.getNomeO + " seu id =
" + idprofissional);
89         }
90         rs.close();
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ProfissionalDAO.java

```
91     stmt.close();
92     c.close();
93     } catch (SQLException ex) {
94         Logger.getLogger(ProfissionalDAO.class.getName()).log(Level.SEVERE, null, ex);
95     }
96     return idprofissional;
97     }
98
99 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/ServicoClassificacaoEstabelecimentoDAO.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.principal;
2
3 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.ServicoClassificacao;
4 import java.sql.Connection;
5 import java.sql.DriverManager;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.sql.Statement;
9 import java.util.logging.Level;
10 import java.util.logging.Logger;
11
12 /**
13  * @author Jucemar
14  */
15 public class ServicoClassificacaoEstabelecimentoDAO {
16
17     private Connection c = null;
18     private Statement stmt = null;
19
20     public ServicoClassificacaoEstabelecimentoDAO() {
21         try {
22             Class.forName("org.sqlite.JDBC");
23             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
24         } catch (Exception e) {
25             //String delete = "DROP TABLE IF EXISTS servico_classificacao_estabelecimento";
26             String tabelaServicoClassificacao = "CREATE TABLE IF NOT EXISTS servico_classificacao_estabelecimento
27 "
28 + "(estabelecimento_cnes INTEGER NOT NULL,"
29 + " classificacao_id INTEGER NOT NULL,"
30 + " PRIMARY KEY(estabelecimento_cnes,classificacao_id),"
31 + " FOREIGN KEY(classificacao_id)"
```

```

31         + " REFERENCES classificacao(id),"
32         + " FOREIGN KEY(estabelecimento_cnes)"
33         + " REFERENCES estabelecimento(cnes));";
34         String index1 = "CREATE INDEX IF NOT EXISTS servico_classificacao_estabelecimento_FKIndex1 ON
servico_classificacao_estabelecimento (classificacao_id);";
35         String index2 = "CREATE INDEX IF NOT EXISTS servico_classificacao_estabelecimento_FKIndex2 ON
servico_classificacao_estabelecimento (estabelecimento_cnes);";
36         String index3 = "CREATE INDEX IF NOT EXISTS servico_classificacao_estabelecimento_PKIndex ON
servico_classificacao_estabelecimento (classificacao_id, estabelecimento_cnes);";
37         stmt = c.createStatement();
38         stmt.executeUpdate(tabelaServicoClassificacao);
39         stmt.executeUpdate(index1);
40         stmt.executeUpdate(index2);
41         stmt.executeUpdate(index3);
42         stmt.close();
43         c.close();
44     } catch (Exception e) {
45         System.err.println(e.getClass().getName() + ": " + e.getMessage());
46     }
47     } catch (ClassNotFoundException ex) {
48         Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
49     } catch (SQLException ex) {
50         Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
51     }
52     }
53
54     public void inserirServicoClassificacao(int cnesEstabelecimento, ServicoClassificacao servClass) {
55         int idClassificacao = getIdClassificacao(servClass);
56         if (idClassificacao != 0) {
57             try {
58                 c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
59             }

```

```
59     this.stmt = this.c.createStatement();
60     String sql = "INSERT INTO servico_classificacao_estabelecimento("
61         + "estabelecimento_cnes, classificacao_id)"
62         + "VALUES("
63         + cnesEstabelecimento + "," +
64         + idClassificacao + ")";
65     System.out.println(sql);
66     this.stmt.executeUpdate(sql);
67     stmt.close();
68     c.close();
69     } catch (SQLException ex) {
70         Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
71     }
72     } else {
73         System.out.println("0 serviço: " + servClass.getServico() + " classificação: " +
servClass.getClassificacao() + " foi descontinuado pelo MS");
74     }
75     }
76
77     private int getIdClassificacao(ServicoClassificacao servClass) {
78         System.out.println("Tentando localizar o Serviço: " + servClass.getServico() + " Classificação: " +
servClass.getClassificacao());
79         int resultado = 0;
80         ResultSet rs = null;
81         int idServico = getIdServico(servClass.getServico());
82         try {
83             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
84             this.stmt = this.c.createStatement();
85             String idClassificacao = "SELECT id FROM classificacao WHERE servico_id=" + idServico + " and codigo=" +
servClass.getClassificacao() + ";";
86             rs = this.stmt.executeQuery(idClassificacao);
```



```
87     if (rs.next()) {
88         resultado = rs.getInt("id");
89     } rs.close();
90     stmt.close();
91     c.close();
92     } catch (SQLException ex) {
93         Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE,
94             null, ex);
95     return resultado;
96 }
97 }
98
99 private int getIdServico(int servico) {
100     int resultado = 0;
101     ResultSet rs = null;
102     try {
103         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
104         this.stmt = this.c.createStatement();
105         String idServico = "SELECT id FROM servico WHERE codigo=" + servico + ";";
106         rs = this.stmt.executeQuery(idServico);
107         if (rs.next()) {
108             resultado = rs.getInt("id");
109         }
110         System.out.println("Codigo do servico: " + servico + " codigo do id: " + resultado);
111         rs.close();
112         stmt.close();
113         c.close();
114     } catch (SQLException ex) {
115         Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
116     }
117     return resultado;
118 }
```

118	}
119	
120	}

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/TabelaEstabelecimento.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.principal;
2
3 /**
4  * @author Jucemar
5  */
6 public final class TabelaEstabelecimento {
7
8     public static final String NOME_TABELA = "estabelecimento";
9
10    public static final class Colunas {
11
12        public static final String CNES = "cnes";
13        public static final String RAZAO_SOCIAL = "razao_social";
14        public static final String NOME_FANTASIA = "nome_fantasia";
15        public static final String TELEFONE = "telefone";
16        public static final String NOME_LOGRADOURO = "nome_logradouro";
17        public static final String NUMERO = "numero";
18        public static final String COMPLEMENTO = "complemento";
19        public static final String BAIRRO = "bairro";
20        public static final String MUNICIPIO = "municipio";
21        public static final String CODIGO_MUNICIPIO = "codigo_municipio";
22        public static final String ESTADO = "estado";
23        public static final String CEP = "cep";
24        public static final String LATITUDE = "latitude";
25        public static final String LONGITUDE = "longitude";
26        public static final String TIPO_ESTABELECIMENTO = "tipo_estabelecimento";
27    }
28 }
```

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.principal;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10
11 /**
12  * @author Jucemar
13  */
14 public class VinculoProfissionalEstabelecimentoDAO {
15
16     private Connection c = null;
17     private Statement stmt = null;
18
19     public VinculoProfissionalEstabelecimentoDAO() {
20         try {
21             Class.forName("org.sqlite.JDBC");
22             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
23             try {
24                 String tabelaProfissional = "CREATE TABLE IF NOT EXISTS vinculo_profissional_estabelecimento "
25                     + "(profissional_id INTEGER NOT NULL,"
26                     + " estabelecimento_cnes INTEGER NOT NULL,"
27                     + " PRIMARY KEY(profissional_id,estabelecimento_cnes),"
28                     + " FOREIGN KEY(profissional_id)"
29                     + " REFERENCES      profissional(id),"
30                     + " FOREIGN KEY(estabelecimento_cnes)"
31                     + " REFERENCES estabelecimento(cnes));";
32             }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar/dimon/jwebcrawlercnes/principal/VinculoProfissionalEstabelecim

```
32      String index1 = "CREATE INDEX IF NOT EXISTS vinculo_profissional_estabelecimento_FKIndex1 ON
vinculo_profissional_estabelecimento (estabelecimento_cnes)";
33      String index2 = "CREATE INDEX IF NOT EXISTS vinculo_profissional_estabelecimento_FKIndex2 ON
vinculo_profissional_estabelecimento (profissional_id)";
34      String index3 = "CREATE INDEX IF NOT EXISTS vinculo_profissional_estabelecimento_PKIndex ON
vinculo_profissional_estabelecimento (profissional_id, estabelecimento_cnes)";
35      stmt = c.createStatement();
36      stmt.executeUpdate(tabelaProfissional);
37      stmt.executeUpdate(index1);
38      stmt.executeUpdate(index2);
39      stmt.executeUpdate(index3);
40      stmt.close();
41      c.close();
42      } catch (Exception e) {
43          System.err.println(e.getClass().getName() + ": " + e.getMessage());
44          System.exit(0);
45      }
46      } catch (ClassNotFoundException ex) {
47          Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
48      } catch (SQLException ex) {
49          Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
50      }
51      }
52
53      public void inserirVinculoDoProfissional(int profissional, int cnes) {
54          boolean inserir = isProfissionalVinculado(profissional, cnes);
55          if (inserir) {
56              try {
57                  c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
58                  this.stmt = this.c.createStatement();
59                  String sql = "INSERT INTO vinculo_profissional_estabelecimento("
59
```

```
60 + "profissional_id, estabelecimento_cnes)"
61 + " VALUES("
62 + profissional + "," + "
63 + cnes + "');"
64 System.out.println(sql);
65 this.stmt.executeUpdate(sql);
66 stmt.close();
67 c.close();
68     } catch (SQLException ex) {
69         Logger.getLogger(EstabelecimentoDAO.class.getName()).log(Level.SEVERE, null, ex);
70     }
71 }
72
73 }
74
75 private boolean isProfissionalVinculado(int profissional, int cnes) {
76     boolean novoVinculo = true;
77     ResultSet rs = null;
78     try {
79         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
80         this.stmt = this.c.createStatement();
81         String buscaVinculoProfissional = "SELECT * FROM vinculo_profissional_estabelecimento WHERE
profissional_id="
82         + profissional
83         + " and estabelecimento_cnes="
84         + cnes
85         + ".";
86         System.out.println(buscaVinculoProfissional);
87         rs = this.stmt.executeQuery(buscaVinculoProfissional);
88         if (rs.next()) {
89             novoVinculo = false;
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/principal/VinculoProfissionalEstabelecim

```
90         System.out.println("Profissional já possui vinculo com este cnes + cbo");
91     }
92     rs.close();
93     stmt.close();
94     c.close();
95     } catch (SQLException ex) {
96         Logger.getLogger(ProfissionalDAO.class.getName()).log(Level.SEVERE, null, ex);
97     }
98     return novoVinculo;
99     }
100
101 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/sigtap/Classificacao.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.sigtap;
2
3 import java.io.Serializable;
4
5 /**
6  * @author Jucemar
7  */
8 public class Classificacao implements Serializable {
9
10     private int codigoServico;
11     private int codigoClassificacao;
12     private String descricao;
13
14     public Classificacao(int codigoServico, int codigoClassificacao, String descricao) {
15         this.codigoServico = codigoServico;
16         this.codigoClassificacao = codigoClassificacao;
17         this.descricao = descricao;
18     }
19
20     public int getCodigoServico() {
21         return codigoServico;
22     }
23
24     public int getCodigoClassificacao() {
25         return codigoClassificacao;
26     }
27
28     public String getDescricao() {
29         return descricao;
30     }
31 }
```


E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/sigtap/ClassificacaoDAO.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.sigtap;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10
11 /**
12  * @author Jucemar
13  */
14 public class ClassificacaoDAO {
15
16     private Connection c = null;
17     private Statement stmt = null;
18
19     public ClassificacaoDAO() {
20         try {
21             Class.forName("org.sqlite.JDBC");
22             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
23             System.out.println("Opened database successfully");
24         }
25         catch (Exception e) {
26             System.out.println("Error: " + e.getMessage());
27         }
28     }
29
30     String delete = "DROP TABLE IF EXISTS classificacao";
31     String tabelaClassificacao = "CREATE TABLE IF NOT EXISTS classificacao "
32         + "(id INTEGER PRIMARY KEY AUTOINCREMENT,"
33         + " servico_id INTEGER NOT NULL,"
34         + " codigo INTEGER NOT NULL,"
35         + " descricao TEXT NOT NULL,"
36         + " FOREIGN KEY(servico_id)"
37         + " REFERENCES servico(id))";
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/sigtap/ClassificacaoDAO.java

```
32     + " REFERENCES servico(id)"
33     + ");";
34     String index = "CREATE INDEX IF NOT EXISTS classificacao_FKIndex ON classificacao (servico_id)";
35     stmt = c.createStatement();
36     stmt.executeUpdate(delete);
37     stmt.executeUpdate(tabelaClassificacao);
38     stmt.executeUpdate(index);
39     stmt.close();
40     c.close();
41 } catch (Exception e) {
42     System.err.println(e.getClass().getName() + ": " + e.getMessage());
43     System.exit(0);
44 }
45 System.out.println("Tabela de classificações criada com sucesso");
46 } catch (ClassNotFoundException ex) {
47     Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);
48 } catch (SQLException ex) {
49     Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);
50 }
51 }
52
53 public void inserirClassificacao(Classificacao classificacao) {
54     ResultSet rs = null;
55     try {
56         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
57         this.stmt = this.c.createStatement();
58
59         String idServico = "SELECT id FROM servico WHERE codigo=" + classificacao.getCodigoServico() + ";";
60         rs = this.stmt.executeQuery(idServico);
61         String sql = "INSERT INTO classificacao("
62             + "codigo, servico_id, descricao)"
```

```

63 + "VALUES("
64 + classificacao.getCodigoClassificacao() + ", "
65 + rs.getInt("id") + ", "
66 + classificacao.getDescricao() + ""
67 + "); ";
68 System.out.println(sql);
69 this.stmt.executeUpdate(sql);
70 rs.close();
71 stmt.close();
72 c.close();
73     catch (SQLException ex) {
74         Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);
75     }
76
77 }
78
79 public void listarClassificacoes() {
80
81     ResultSet rs;
82     try {
83         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
84         this.stmt = this.c.createStatement();
85         rs = this.stmt.executeQuery("SELECT * FROM classificacao");
86
87         while (rs.next()) {
88             System.out.print("serviço: " + rs.getString("id") + " ");
89             System.out.print("Classificação: " + rs.getString("servico_id") + " ");
90             System.out.print("Descrição: " + rs.getString("descricao"));
91             System.out.println(" ");
92         }
93         rs.close();
94     }

```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/sigtap/ClassificacaoDAO.java

```
94     stmt.close();
95     c.close();
96
97     } catch (SQLException e) {
98         e.printStackTrace();
99     }
100 }
101
102 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/sigtap/LeitorDeArquivos.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.sigtap;
2
3 import java.io.BufferedReader;
4 import java.io.FileInputStream;
5 import java.io.FileNotFoundException;
6 import java.io.IOException;
7 import java.io.InputStreamReader;
8 import java.io.UnsupportedEncodingException;
9 import java.util.ArrayList;
10 import java.util.Collections;
11 import java.util.Comparator;
12
13 /**
14  * @author Jucemar
15  */
16 public class LeitorDeArquivos {
17
18     public ArrayList<Servico> carregarArquivoTxtServicos() throws FileNotFoundException, IOException {
19         ArrayList<Servico> resultado = new ArrayList<Servico>();
20         String nomeArquivo = "tb_servico.txt";
21         BufferedReader lerArq = new BufferedReader(new InputStreamReader(new FileInputStream(nomeArquivo),
22             "ISO-8859-1"));
23         String linha = "";
24         while ((linha = lerArq.readLine()) != null) {
25             String res = limpar(linha);
26             String codigo = res.substring(0, 3);
27             String descricao = res.substring(3, res.length());
28             Servico s = new Servico(Integer.parseInt(codigo), descricao);
29             resultado.add(s);
30         }
31         lerArq.close();
32     }
33 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/sigtap/LeitorDeArquivos.java

```
31 lerArq.close();
32 return resultado;
33 }
34
35 public ArrayList<Classificacao> carregarArquivoTxtClassificacoes() throws FileNotFoundException,
UnsupportedEncodingException, IOException {
36     ArrayList<Classificacao> resultado = new ArrayList<Classificacao>();
37     String nomeArquivo = "tb_servico_classificacao.txt";
38     BufferedReader lerArq = new BufferedReader(new InputStreamReader(new FileInputStream(nomeArquivo),
"ISO-8859-1"));
39     String linha = "";
40     while ((linha = lerArq.readLine()) != null) {
41         String res = limpar(linha);
42         String servico = res.substring(0, 3); classificacao
43         String = res.substring(3, 6); descricao =
44         String res.substring(6, res.length());
45         Classificacao c = new Classificacao(Integer.parseInt(servico), Integer.parseInt(classificacao),
descricao);
46         resultado.add(c);
47     }
48     lerArq.close();
49     lerArq.close();
50     Collections.sort(resultado, new Comparator() {
51         public int compare(Object o1, Object o2) {
52             Classificacao p1 = (Classificacao) o1;
53             Classificacao p2 = (Classificacao) o2;
54             return p1.getCodigoServico() < p2.getCodigoServico() ? -1 : (p1.getCodigoServico() >
p2.getCodigoServico() ? +1 : 0);
55         }
56     });
57     return resultado;
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/sigtap/LeitorDeArquivos.java

```
58     }
59
60     private String limpar(String txt) {
61         String resultado = null;
62         int i = 0;
63         boolean fim = false;
64         while (fim == false) {
65             int n = txt.indexOf(" ");
66             if (n > -1) {
67                 fim = true;
68                 resultado = txt.substring(0, n);
69             }
70             i++;
71         }
72         return resultado;
73     }
74
75 }
```


E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/sigtap/Servico.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.sigtap;
2
3 import java.io.Serializable;
4
5 /**
6  * @author Jucemar
7  */
8 public class Servico implements Serializable {
9
10     private int codigo;
11     private String descricao;
12
13     public Servico(int codigo, String descricao) {
14         this.codigo = codigo;
15         this.descricao = descricao;
16     }
17
18     public int getCodigo() {
19         return codigo;
20     }
21
22     public String getDescricao() {
23         return descricao;
24     }
25 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/sigtap/ServicoDAO.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.sigtap;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10
11 /**
12  * @author Jucemar
13  */
14 public class ServicoDAO {
15
16     private Connection c = null;
17     private Statement stmt = null;
18
19     public ServicoDAO() {
20         try {
21             Class.forName("org.sqlite.JDBC");
22             c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
23             System.out.println("Opened database successfully");
24         }
25         catch (Exception e) {
26             String deleta = "DROP TABLE IF EXISTS servico";
27             String tabelaServico = "CREATE TABLE IF NOT EXISTS servico "
28                 + "(id INTEGER PRIMARY KEY AUTOINCREMENT,"
29                 + "    codigo INTEGER NOT NULL,"
30                 + "    descricao TEXT NOT NULL"
31                 + ");";
32             String index = "CREATE INDEX IF NOT EXISTS servico_PKIndex ON servico (id);";
33         }
34     }
35 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/sigtap/ServicoDAO.java

```
32 stmt = c.createStatement();
33 stmt.executeUpdate(deleta);
34 stmt.executeUpdate(tabelaServico);
35 stmt.executeUpdate(index);
36 stmt.close();
37 c.close();
38 } catch (Exception e) {
39     System.err.println(e.getClass().getName() + ": " + e.getMessage());
40     System.exit(0);
41 }
42 System.out.println("Tabela de servios criada com sucesso");
43 } catch (ClassNotFoundException ex) {
44     Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);
45 } catch (SQLException ex) {
46     Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);
47 }
48 }
49
50 public void inserirServico(Servico servico) {
51     try {
52         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
53         this.stmt = this.c.createStatement();
54         try {
55             String sql = "INSERT INTO servico("
56                 + "codigo, descricao)"
57                 + " VALUES("
58                 + servico.getCodigo() + ", "
59                 + servico.getDescricao() + " "
60                 + ");";
61             System.out.println(sql);
62             this.stmt.executeUpdate(sql);
63         }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/sigtap/ServicoDAO.java

```
63     } catch (SQLException e) {
64         e.printStackTrace();
65     }
66     stmt.close();
67     c.close();
68     } catch (SQLException ex) {
69         Logger.getLogger(ClassificacaoDAO.class.getName()).log(Level.SEVERE, null, ex);
70     }
71 }
72
73 public void listarServicos() {
74     ResultSet rs;
75     try {
76         c = DriverManager.getConnection("jdbc:sqlite:estabelecimentos.db");
77         this.stmt = this.c.createStatement();
78         rs = this.stmt.executeQuery("SELECT * FROM servico");
79         while (rs.next()) {
80             System.out.print("serviço: " + rs.getString("id") + " ");
81             System.out.print("Descrição: " + rs.getString("descricao"));
82             System.out.println(" ");
83         } rs.close();
84         stmt.close();
85         c.close();
86     } catch (SQLException e) {
87         e.printStackTrace();
88     }
89 }
90
91 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/sigtap/SigTap.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.sigtap;
2
3 import java.io.FileNotFoundException;
4 import java.io.IOException;
5 import java.util.ArrayList;
6
7 /**
8  * @author Jucemar
9  */
10 public class SigTap {
11
12     public static void iniciar() throws FileNotFoundException, IOException {
13         LeitorDeArquivos leitor = new LeitorDeArquivos(); ArrayList<Servico> s =
14         leitor.carregarArquivoTxtServicos(); ArrayList<Classificacao> c =
15         leitor.carregarArquivoTxtClassificacoes(); ServicoDAO sd = new
16         ServicoDAO();
17         for (Servico ser : s) {
18             sd.inserirServico(ser);
19         }
20         ClassificacaoDAO cd = new ClassificacaoDAO();
21         for (Classificacao cla : c) {
22             cd.inserirClassificacao(cla);
23         }
24         sd.listarServicos();
25         cd.listarClassificacoes();
26     }
27 }
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/utilitarios/ManipuladorDeArquivos.java

```
1 package br.com.jucemar_dimon.jwebcrawlercnes.utilitarios;
2
3 import br.com.jucemar_dimon.jwebcrawlercnes.entidades.Municipio;
4 import java.io.BufferedReader;
5 import java.io.BufferedOutputStream;
6 import java.io.File;
7 import java.io.FileInputStream;
8 import java.io.FileOutputStream;
9 import java.io.IOException;
10 import java.io.ObjectInputStream;
11 import java.io.ObjectOutputStream;
12 import java.util.ArrayList;
13 import java.util.logging.Level;
14 import java.util.logging.Logger;
15
16 /**
17  * @author Jucemar
18  */
19 public class ManipuladorDeArquivos {
20
21     private File arquivo;
22
23     public ManipuladorDeArquivos() {
24         arquivo = new File("estabelecimentos_temp");
25         if (arquivo.exists() == false) {
26             try {
27                 arquivo.createNewFile();
28                 System.out.println("Um arquivo temporários de estabelecimentos foi criado com sucesso");
29             } catch (IOException ex) {
30                 Logger.getLogger(ManipuladorDeArquivos.class.getName()).log(Level.SEVERE, null, ex);
31             }
32         }
33     }
34 }
```

```
32     }
33 }
34
35 public void salvarDadosEmArquivoTemporario(ArrayList<Municipio> dados) throws IOException {
36     ObjectOutputStream outputStream = null;
37     try {
38         outputStream = new ObjectOutputStream(new BufferedOutputStream(new FileOutputStream(arquivo)));
39         outputStream.writeObject(dados);
40     } catch (IOException ex) {
41         Logger.getLogger(ManipuladorDeArquivos.class.getName()).log(Level.SEVERE, null, ex);
42     } finally {
43         if (outputStream != null) {
44             outputStream.close();
45         }
46     }
47 }
48
49 public ArrayList<Municipio> leiaDadosDEArquivoTemporario() throws IOException {
50     ArrayList<Municipio> dados = null;
51     ObjectInputStream inputStream = null;
52     try {
53         inputStream = new ObjectInputStream(new BufferedInputStream(
54             new FileInputStream(arquivo)));
55         dados = (ArrayList<Municipio>) inputStream.readObject();
56         System.out.println("Há registros de um processamento anterior, a captura será retomado do ponto onde
57 parou");
58     } catch (IOException ex) {
59         System.out.println("Não há registros de processamentos anteriores, a captura será feita desde o início");
60     } catch (ClassNotFoundException ex) {
61         Logger.getLogger(ManipuladorDeArquivos.class.getName()).log(Level.SEVERE, null, ex);
62     } finally {
```

E:/Projetos Netbeans/JWebCrawlerCNES/src/br/com/jucemar_dimon/jwebcrawlercnes/utilitarios/ManipuladorDeArquivos.java

```
62         if (inputStream != null) {
63             inputStream.close();
64         }
65     }
66     return dados;
67 }
68
69 }
```


APÊNDICE C - CODIGO FONTE DO APLICATIVO ANDROID

File - E:\Android\Projetos\CadEsaude\app\src\main\AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="br.com.jucemar_dimon.cadesaude"
4     android:versionCode="1"
5     android:versionName="0.01">
6
7     <uses-feature
8         android:name="android.hardware.screen.portrait"
9         android:required="true" />
10
11     <uses-feature
12         android:name="android.hardware.telephony"
13         android:required="true" />
14
15     <uses-feature
16         android:name="android.hardware.location.gps"
17         android:required="true" />
18
19     <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
20
21     <application
22         android:allowBackup="true"
23         android:icon="@mipmap/ic_launcher"
24         android:label="@string/app_name"
25         android:screenOrientation="portrait"
26         android:supportRtl="true"
27         android:theme="@style/MeuTemaMaterial">
28         <activity
29             android:name=".MainActivity"
30             android:label="@string/app_name"
31             android:launchMode="singleTop"
32             android:screenOrientation="portrait"
33             android:windowSoftInputMode="adjustPan|adjustNothing">
34             <intent-filter>
35                 <action android:name="android.intent.action.VIEW" />
36             </intent-filter>
37             <intent-filter>
38                 <action android:name="android.intent.action.VIEW" />
39             </intent-filter>
40         </activity>
41     </application>
42 </manifest>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\AndroidManifest.xml

```
36     <action android:name="android.intent.action.SEARCH" />
37     </intent-filter>
38     <meta-data
39         android:name="android.app.searchable"
40         android:resource="@xml/searchable" />
41     </activity>
42     <activity
43         android:name=".PreferenciasActivity"
44         android:label="@string/titulo_config"
45         android:launchMode="singleTop"
46         android:screenOrientation="portrait"
47         android:theme="@style/MeuTemaMaterial.Prefencias" />
48     <activity
49         android:name=".AjudaActivity"
50         android:label="@string/titulo_ajuda"
51         android:launchMode="singleTop"
52         android:screenOrientation="portrait" />
53     <activity
54         android:name=".SobreActivity"
55         android:label="@string/titulo_sobre"
56         android:launchMode="singleTop"
57         android:screenOrientation="portrait" />
58     <activity
59         android:name=".BuscaGPSActivity"
60         android:label="@string/titulo_busca_gps_activity"
61         android:launchMode="singleTop"
62         android:screenOrientation="portrait"
63         android:windowSoftInputMode="adjustPan|adjustNothing">
64     <intent-filter>
65         <action android:name="android.intent.action.SEARCH" />
66     </intent-filter>
67     <meta-data
68         android:name="android.app.searchable"
69         android:resource="@xml/searchable" />
70     --
```

File - E:\Android\Projetos\CadEsaude\app\src\main\AndroidManifest.xml

```
71 </activity>
72 <activity            android:name=".EstabelecimentoActivity"
73     android:label="@string/title_activity_estabelecimento"
74     android:launchMode="singleTop"
75     android:screenOrientation="portrait" />
76 <meta-data          android:name="com.google.android.gms.version"
77     android:value="@integer/google_play_services_version" />
78 <activity
79     android:name=".BuscaGeralActivity"
80     android:label="@string/title_activity_busca_geral"
81     android:launchMode="singleTop"
82     android:screenOrientation="portrait" />
83 <activity
84     android:name=".PrimeirosPassosActivity"
85     android:label="@string/title_activity_primeiros_passos"
86     android:launchMode="singleTop"
87     android:screenOrientation="portrait"
88     android:theme="@style/FullscreenTheme" />
89 <activity            android:name=".PrimeiraConfiguracaoActivity"
90     android:label="@string/titulo_config"
91     android:launchMode="singleTop"
92     android:screenOrientation="portrait"
93     android:theme="@style/MeuTemaMaterial.Prefencias" />
94 <activity            android:name=".AjudaTiposEstabelecimentosActivity"
95     android:label="@string/titulo_actvt_ajuda_estabelecimentos"
96     android:launchMode="singleTop"
97     android:screenOrientation="portrait" />
98 <activity            android:name=".AjudaTiposDeServicosActivity"
99     android:label="@string/titulo_actvt_ajuda_servicos"
100    android:launchMode="singleTop"
101
102
103
104 ...
```

File - E:\Android\Projetos\CadEsaude\app\src\main\AndroidManifest.xml

```
106     android:screenOrientation="portrait" />
107     <activity
108         android:name=".TelaInicialActivity"
109         android:launchMode="singleTop"
110         android:screenOrientation="portrait">
111         <intent-filter>
112             <action android:name="android.intent.action.MAIN" />
113             <category android:name="android.intent.category.LAUNCHER" />
114         </intent-filter>
115     </activity>
116 </application>
117
118 </manifest>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\AjudaActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3  * Created by Jucemar on 10/03/2016.
4  */
5
6 import android.content.Intent;
7 import android.support.v7.app.AppCompatActivity;
8 import android.os.Bundle;
9 import android.support.v7.widget.Toolbar;
10 import android.view.MenuItem;
11 import android.view.View;
12 import android.widget.Button;
13
14 public class AjudaActivity extends AppCompatActivity implements View.OnClickListener {
15
16     private Toolbar toolbar;
17     private Button buttonTiposEstabelecimentos;
18     private Button buttonTiposServicos;
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_ajuda);
24         toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_ajuda);
25         setSupportActionBar(toolbar);
26         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
27         toolbar.setTitle("Ajuda");
28         buttonTiposEstabelecimentos = (Button) findViewById(R.id.btn_tipos_estabelecimentos);
29         buttonTiposEstabelecimentos.setOnClickListener(this);
30         buttonTiposServicos = (Button) findViewById(R.id.btn_tipos_servicos);
31         buttonTiposServicos.setOnClickListener(this);
32     }
33
34     @Override
35     public boolean onOptionsItemSelected(MenuItem item) {
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\AjudaActivity.java

```
36  int id = item.getItemId();
37  switch (id) {
38      case android.R.id.home:
39      onBackPressed();
40      break;
41  }
42  return super.onOptionsItemSelected(item);
43  }
44
45  @Override
46  public void onClick(View v) {
47      Intent i = null;
48      switch (v.getId()) {
49          case R.id.btn_tipos_estabelecimentos:
50              i = new Intent(this, AjudaTiposEstabelecimentosActivity.class);
51              break;
52          case R.id.btn_tipos_servicos:
53              i = new Intent(this, AjudaTiposDeServicosActivity.class);
54              break;
55          }
56      startActivity(i);
57  }
58  }
59  }
60
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juceamar_dimon\cadesaude\AjudaTiposDeServicosActivity.java

```
1 package br.com.juceamar_dimon.cadesaude;
2 /**
3  * Created by Juceamar on 10/03/2016.
4  */
5
6 import android.os.Bundle;
7 import android.support.v7.app.AppCompatActivity;
8 import android.support.v7.widget.Toolbar;
9 import android.view.MenuItem;
10
11 public class AjudaTiposDeServicosActivity extends AppCompatActivity {
12     private Toolbar toolbar;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_ajuda_tipos_de_servicos);
18         toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_ajuda_tipos_servicos);
19         setSupportActionBar(toolbar);
20         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
21         toolbar.setTitle("Tipos de serviços");
22     }
23
24     @Override
25     public boolean onOptionsItemSelected(MenuItem item) {
26         int id = item.getItemId();
27         switch (id) {
28             case android.R.id.home:
29                 onBackPressed();
30                 break;
31         }
32         return super.onOptionsItemSelected(item);
33     }
34 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juceamar_dimon\cadesaude\AjudaTiposEstabelecimentosActivity.java

```
1 package br.com.juceamar_dimon.cadesaude;
2 /**
3  * Created by Juceamar on 10/03/2016.
4  */
5
6 import android.os.Bundle;
7 import android.support.v7.app.AppCompatActivity;
8 import android.support.v7.widget.Toolbar;
9 import android.view.MenuItem;
10
11 public class AjudaTiposEstabelecimentosActivity extends AppCompatActivity {
12     private Toolbar toolbar;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_ajuda_tipos_estabelecimentos);
18         toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_ajuda_tipos_estabelecimentos);
19         setSupportActionBar(toolbar);
20         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
21         toolbar.setTitle("Tipos de estabelecimentos");
22     }
23
24     @Override
25     public boolean onOptionsItemSelected(MenuItem item) {
26         int id = item.getItemId();
27         switch (id) {
28             case android.R.id.home:
29                 onBackPressed();
30                 break;
31         }
32         return super.onOptionsItemSelected(item);
33     }
34 }
```


File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\juceamar_dimon\cadesaude\Atendimento.java

```
1 package br.com.juceamar_dimon.cadesaude;
2
3 import android.os.Parcel;
4 import android.os.Parcelable;
5
6 /**
7  * @author Juceamar
8  */
9 public class Atendimento implements Parcelable {
10     private String tipoAtendimento;
11     private String convenio;
12     private String cnes;
13
14     public Atendimento(String cnes, String tipoAtendimento, String convenio) {
15         this.tipoAtendimento = tipoAtendimento;
16         this.convenio = convenio;
17         this.cnes = cnes;
18     }
19
20     protected Atendimento(Parcel in) {
21         tipoAtendimento = in.readString();
22         convenio = in.readString();
23         cnes = in.readString();
24     }
25
26     public static final Creator<Atendimento> CREATOR = new Creator<Atendimento>() {
27         @Override
28         public Atendimento createFromParcel(Parcel in) {
29             return new Atendimento(in);
30         }
31
32         @Override
33         public Atendimento[] newArray(int size) {
34             return new Atendimento[size];
35         }
36     }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Atendimento.java

```
36     };
37
38
39     @Override
40     public int describeContents() {
41         return 0;
42     }
43
44     @Override
45     public void writeToParcel(Parcel dest, int flags) {
46         dest.writeString(tipoAtendimento);
47         dest.writeString(convenio);
48         dest.writeString(cnes);
49     }
50
51     public String getTipoAtendimento() {
52         return tipoAtendimento;
53     }
54
55     public void setTipoAtendimento(String tipoAtendimento) {
56         this.tipoAtendimento = tipoAtendimento;
57     }
58
59     public String getConvenio() {
60         return convenio;
61     }
62
63     public void setConvenio(String convenio) {
64         this.convenio = convenio;
65     }
66
67     public String getCnes() {
68         return cnes;
69     }
70 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Atendimento.java

```
71 public void setCnes(String cnes) {  
72     this.cnes = cnes;  
73 }  
74  
75 }  
76
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juccemar_dimon\cadesaude\AtendimentoAdapter.java

```
1 package br.com.juccemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.AdapterView;
8 import android.widget.AdapterView.OnItemClickListener;
9
10 import java.util.ArrayList;
11
12 /**
13  * Created by Juccemar on 19/03/2016.
14  */
15 public class AtendimentoAdapter extends ArrayAdapter<Atendimento> {
16
17
18     public AtendimentoAdapter(Context context, ArrayList<Atendimento> objects) {
19         super(context, R.layout.fragment_listas, objects);
20     }
21
22     @Override
23     public View getView(int position, View convertView, ViewGroup parent) {
24         Atendimento a = getItem(position);
25         if (convertView == null) {
26             convertView = LayoutInflater.from(getContext()).inflate(R.layout.item_listas, parent, false);
27
28             TextView tvAtendimento = (TextView) convertView.findViewById(R.id.item_linha_1);
29             TextView tvConvenio = (TextView) convertView.findViewById(R.id.item_linha_2);
30             tvAtendimento.setText(a.getTipoAtendimento());
31             tvConvenio.setText(a.getConvenio());
32             return convertView;
33         }
34     }
35 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juceamar_dimon\cadesaude\AtendimentoDAO.java

```
1 package br.com.juceamar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.util.Log;
7
8 import java.util.ArrayList;
9 import java.util.Set;
10
11 /**
12  * Created by Juceamar on 19/03/2016.
13  */
14
15 public class AtendimentoDAO {
16
17     private static AtendimentoDAO instancia;
18     private SQLiteDatabase bd;
19     private String[] columnas;
20
21     public static AtendimentoDAO getInstance(Context contexto) {
22         if (instancia == null) {
23             instancia = new AtendimentoDAO(contexto);
24         }
25         return instancia;
26     }
27
28     private AtendimentoDAO(Context contexto) {
29         BancoDeDados bancoDeDados = BancoDeDados.getInstance(contexto);
30         bd = bancoDeDados.getWritableDatabase();
31         columnas = new String[] {
32             TabelaAtendimento.Columnas.ATENDIMENTO,
33             TabelaAtendimento.Columnas.CONVENIO,
34             TabelaAtendimento.Columnas.ESTABELECIMENTO_CNES,
35             TabelaAtendimento.Columnas.ID
36         };
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\AtendimentoDAO.java

```
36     };
37 }
38
39 public ArrayList<Atendimento> listarAtendimentos(String cnes) {
40     String[] argumentos = {cnes};
41     Cursor c = bd.query(TabelaAtendimento.NOME_TABELA, colunas, TabelaAtendimento.Colunas.ESTABELECIMENTO_CNES +
42         "=?", argumentos, null, null, TabelaAtendimento.Colunas.ATENDIMENTO);
43     Log.i("Cursor",
44         String.valueOf(c.getCount()));
45     ArrayList<Atendimento> resultado = new ArrayList<Atendimento>();
46     if (c.moveToFirst()) {
47         do {
48             Atendimento a = extrairDoCursor(c);
49             resultado.add(a);
50         } while (c.moveToNext());
51     }
52     c.close();
53     Log.e("ResultadobuscaProf", String.valueOf(resultado.size()));
54     return resultado;
55 }
56
57 public static final Atendimento extrairDoCursor(Cursor c) {
58     String atendimento = c.getString(c.getColumnIndex(TabelaAtendimento.Colunas.ATENDIMENTO));
59     String convenio = c.getString(c.getColumnIndex(TabelaAtendimento.Colunas.CONVENIO));
60     String cnes = String.valueOf(c.getInt(c.getColumnIndex(TabelaAtendimento.Colunas.ESTABELECIMENTO_CNES)));
61     return new Atendimento(cnes, atendimento, convenio);
62 }
63
64 public CharSequence[] listarTiposAtendimento(Set<String> setMunicipios) {
65     CharSequence[] resultado = null;
66     if (setMunicipios.size() > 0) {
67         StringBuffer filtro = new StringBuffer();
68         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
69         for (int i = 0; i < municipios.length; i++) {
70             filtro.append(TabelaEstabelecimento.NOME_TABELA + ". " + TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO
71                 + "=? ");
72         }
73     }
74 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\AtendimentoDAO.java

```
69         if (i < municipios.length - 1) {
70             filtro.append("OR ");
71         }
72     }
73     String query = "SELECT DISTINCT atendimento_estabelecimento.atendimento FROM estabelecimento JOIN
    atendimento_estabelecimento ON estabelecimento.cnes=atendimento_estabelecimento.estabelecimento_cnes WHERE " + filtro
    + " ORDER BY atendimento_estabelecimento.atendimento";
74     Cursor c = bd.rawQuery(query, municipios);
75     resultado = new CharSequence[c.getCount()];
76     int i = 0;
77     if (c.moveToFirst()) {
78         do {
79             resultado[i] = c.getString(c.getColumnIndex(TabelaAtendimento.Colunas.ATENDIMENTO));
80             i++;
81         } while (c.moveToNext());
82     }
83     c.close();
84 }
85 return resultado;
86 }
87
88 public CharSequence[] listarTiposConvenios(Set<String> setMunicipios) {
89     CharSequence[] resultado = null;
90     if (setMunicipios.size() > 0) {
91         StringBuffer filtro = new StringBuffer();
92         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
93         for (int i = 0; i < municipios.length; i++) {
94             filtro.append(TabelaEstabelecimento.NOME_TABELA + "." + TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO
    + "=? ");
95         }
96         if (i < municipios.length - 1) {
97             filtro.append("OR ");
98         }
99     }
    String query = "SELECT DISTINCT atendimento_estabelecimento.convenio FROM estabelecimento JOIN
    atendimento_estabelecimento ON estabelecimento.cnes=atendimento_estabelecimento.estabelecimento_cnes WHERE " + filtro
```


File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\AtendimentoDAO.java

```
132     Estabelecimento e = EstabelecimentoDAO.extrairDoCursor(c);
133     resultado.add(e);
134     } while (c.moveToNext());
135 }
136 c.close();
137 return resultado;
138 }
139
140 public ArrayList<Estabelecimento> buscaStringEmAtendimentos(Set<String> setMunicipios, String sentenca) {
141     StringBuffer filtroMunicipios = new StringBuffer();
142     String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
143     for (int i = 0; i < municipios.length; i++) {
144         filtroMunicipios.append("'" + municipios[i] + "'");
145         if (i < municipios.length - 1) {
146             filtroMunicipios.append(", ");
147         }
148     }
149     String query = "SELECT * FROM estabelecimento WHERE cnes IN (\n" +
150         "SELECT DISTINCT atendimento_estabelecimento_estabelecimento_cnes FROM atendimento_estabelecimento
151         JOIN " +
152         "estabelecimento ON estabelecimento.cnes=atendimento_estabelecimento_estabelecimento_cnes WHERE
153         atendimento_estabelecimento.atendimento " +
154         "LIKE '%" + sentenca + "%' AND estabelecimento.codigo_municipio IN (" + filtroMunicipios + "))";
155     Log.e("Query", query);
156     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
157     Cursor c = bd.rawQuery(query, null);
158     if (c.moveToFirst()) {
159         do {
160             Estabelecimento e = EstabelecimentoDAO.extrairDoCursor(c);
161             resultado.add(e);
162         } while (c.moveToNext());
163     }
164     c.close();
165     return resultado;
166 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\AtendimentoDAO.java

```
165
166 }
167
168
169
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\AtendimentoToFragment.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4  * Created by Jucemar on 18/03/2016.
5  */
6
7 import android.os.Bundle;
8 import android.support.v4.app.Fragment;
9 import android.view.LayoutInflater;
10 import android.view.View;
11 import android.view.ViewGroup;
12 import android.widget.LinearLayout;
13 import android.widget.ListView;
14 import android.widget.TextView;
15
16 import java.util.ArrayList;
17
18 public class AtendimentoToFragment extends Fragment {
19
20     public AtendimentoToFragment() {
21     }
22
23     @Override
24     public void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26     }
27
28     @Override
29     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
30         View v = inflater.inflate(R.layout.fragment_listas, container, false);
31         ArrayList<Atendimento> listaDeAtendimentos = getArguments().getParcelableArrayList("listaDeAtendimentos");
32         if (listaDeAtendimentos.size() > 0) {
33             ListView lv = (ListView) v.findViewById(R.id.list_view);
34             lv.setDivider(null);
35             AtendimentoAdapter aa = new AtendimentoAdapter(getActivity(), listaDeAtendimentos);
36         }
37     }
38 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\AtendimentoFfragment.java

```
36     lv.setAdapter(aa);
37     } else {
38         LinearLayout layout = (LinearLayout) v.findViewById(R.id.view_vazio);
39         LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(LinearLayout.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT);
40         params.bottomMargin = 16;
41         TextView textView = new TextView(getActivity());
42         textView.setLayoutParams(params);
43         textView.setText("Não há informações cadastradas");
44         layout.addView(textView);
45     }
46     return v;
47 }
48
49 }
50
51
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BancoDeDados.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.database.sqlite.SQLiteDatabase;
5 import android.database.sqlite.SQLiteOpenHelper;
6 import android.util.Log;
7
8 import java.io.File;
9 import java.io.FileOutputStream;
10 import java.io.IOException;
11 import java.io.InputStream;
12 import java.io.OutputStream;
13
14 /**
15  * Created by Jucemar on 10/03/2016.
16  */
17 public class BancoDeDados extends SQLiteOpenHelper {
18     public static String BD_CAMINHO = "/data/data/br.com.jucemar_dimon.cadesaude/databases/";
19     public static String BD_NOME = "estabelecimentos.db";
20     public static final int BD_VERSAO = 201610;
21     private static BancoDeDados instancia;
22     private Context contexto;
23
24     public static BancoDeDados getInstancia(Context contexto) {
25         if (instancia == null) {
26             instancia = new BancoDeDados(contexto);
27             String myPath = BD_CAMINHO + BD_NOME;
28             File banco = new File(myPath);
29             if (banco.exists() && !banco.isDirectory()) {
30                 banco.delete();
31             }
32             String caminho = "/data/data/br.com.jucemar_dimon.cadesaude/databases/";
33             File pasta = new File(caminho);
34             pasta.mkdirs();
35             File bancoPronto = new File(pasta, BD_NOME);
36         }
37     }
38 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BancoDeDados.java

```
36     try {
37         bancoPronto.createNewFile();
38         Log.e("BancoDeDados", "arquivo de banco de dados não existe, ele será copiado");
39         instancia.copiarBancoDeDados();
40     } catch (IOException e) {
41         Log.e("BancoDeDados", "Houve algum erro durante a criação ou cópia do banco de dados");
42     }
43 }
44 return instancia;
45 }
46
47 private BancoDeDados(Context contexto) {
48     super(contexto, BD_NOME, null, BD_VERSAO);
49     this.contexto = contexto;
50 }
51
52 @Override
53 public void onCreate(SQLiteDatabase db) {
54     //Não implementado
55 }
56
57 public void copiarBancoDeDados() {
58     try {
59         InputStream fluxoEntrada = contexto.getAssets().open(BD_NOME);
60         String arquivoFinal = BD_CAMINHO + BD_NOME;
61         OutputStream fluxoSaida = new FileOutputStream(arquivoFinal);
62         byte[] buffer = new byte[1024];
63         int tamanho;
64         while ((tamanho = fluxoEntrada.read(buffer)) > 0) {
65             fluxoSaida.write(buffer, 0, tamanho);
66             fluxoSaida.flush();
67             fluxoEntrada.close();
68             fluxoSaida.close();
69         }
70     } catch (IOException e) {
71         Log.e("BancoDeDados", "O Arquivo de banco de dados foi copiado com sucesso ");
72     }
73 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BancoDeDados.java

```
71     } catch (Exception e) {
72         Log.e("BancoDeDados", "Erro " + e.getMessage());
73     }
74 }
75
76 @Override
77 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
78     //Não implementado
79 }
80
81 }
82
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGeralActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3  * Created by Jucemar on 10/03/2016.
4  */
5
6 import android.content.Context;
7 import android.content.SharedPreferences;
8 import android.os.AsyncTask;
9 import android.os.Bundle;
10 import android.preference.PreferenceManager;
11 import android.support.design.widget.CoordinatorLayout;
12 import android.support.design.widget.Snackbar;
13 import android.support.v7.app.AppCompatActivity;
14 import android.support.v7.widget.DefaultItemAnimator;
15 import android.support.v7.widget.LinearLayoutManager;
16 import android.support.v7.widget.RecyclerView;
17 import android.support.v7.widget.Toolbar;
18 import android.util.Log;
19 import android.view.MenuItem;
20 import android.view.View;
21 import android.widget.ImageView;
22 import android.widget.ProgressBar;
23 import android.widget.TextView;
24
25 import com.github.clans.fab.FloatingActionButton;
26
27 import java.util.ArrayList;
28 import java.util.Arrays;
29 import java.util.Collections;
30 import java.util.HashSet;
31 import java.util.Set;
32
33 public class BuscaGeralActivity extends AppCompatActivity implements InterfaceBuscaGeralPorString, View.OnClickListener
34 {
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGeralActivity.java

```
35 private ArrayList<Estabelecimento> estabelecimentos;
36 private Set<String> listaDeMunicipios;
37 private EstabelecimentoDAO estabDAO;
38 private ProgressBar barraDeProgresso;
39 private RecyclerView meuRecyclerView;
40 private EstabelecimentoAdapter meuAdapter;
41 private RecyclerView.LayoutManager meuLayoutManager;
42 private CarregadorDeDados carregadorDeDados;
43 private int posicaoNaLista; Set<String>
44 private listaDeFavoritos; SharedPreferences config;
45 private ArrayList<Profissional> listaDeProfissionais;
46 private ArrayList<Atendimento> listaDeAtendimento;
47 private ArrayList<ServicoClassificacao> listaDeServicos;
48 private ImageView imagemBackground;
49 private TextView txtBackground;
50 private TextView txtBackgroundMunicipios;
51 private FloatingActionButton fab;
52 private SharedPreferences preferencias;
53 private ArrayList<String> codMunicipio;
54 private ArrayList<String> nomeMunicipio;
55
56
57 @Override
58 protected void onCreate(Bundle savedInstanceState) {
59     super.onCreate(savedInstanceState);
60     setContentView(R.layout.activity_busca_geral);
61     barraDeProgresso = (ProgressBar) findViewById(R.id.barra_progresso_busca_geral_activity);
62     barraDeProgresso.setVisibility(View.GONE);
63     Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_busca_geral);
64     setSupportActionBar(toolbar);
65     getSupportActionBar().setDisplayHomeAsUpEnabled(true);
66     imagemBackground = (ImageView) findViewById(R.id.background_erro_busca_geral);
67     txtBackground = (TextView) findViewById(R.id.label_erro_busca);
68     txtBackground.setText(R.string.titulo_dialog_buscar_geral);
69     txtBackgroundMunicipios = (TextView) findViewById(R.id.label_municipios para busca);
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGeralActivity.java

```
70 fab = (FloatingActionButton) findViewById(R.id.fab_activity_busca_geral);
71 fab.setOnClickListener(this);
72 estabelecimentos = new ArrayList<Estabelecimento>();
73 meuRecyclerView = (RecyclerView) findViewById(R.id.lista_estabelecimentos);
74 meuRecyclerView.setVisibility(View.GONE); meuLayoutManager =
75     new LinearLayoutManager(this);
76 meuRecyclerView.setLayoutManager(meuLayoutManager);
77 meuRecyclerView.setItemAnimator(new DefaultItemAnimator());
78 meuRecyclerView.setHasFixedSize(true);
79 mostrarMunicipiosEscolhidos();
80 }
81
82 @Override
83 public boolean onOptionsItemSelected(MenuItem item) {
84     int id = item.getItemId();
85     switch (id) {
86         case android.R.id.home:
87             onBackPressed();
88             break;
89     }
90     return super.onOptionsItemSelected();
91 }
92
93 private void mostrarMunicipiosEscolhidos() {
94     preferencias = PreferenceManager.getDefaultSharedPreferences(this);
95     String[] arrayNomes = getResources().getStringArray(R.array.nomes_municipios_view);
96     nomeMunicipio = new ArrayList<String>(Arrays.asList(arrayNomes));
97     String[] arrayCodigos = getResources().getStringArray(R.array.codigos_municipios);
98     codMunicipio = new ArrayList<String>(Arrays.asList(arrayCodigos));
99     Set<String> temp = preferencias.getStringSet(getResources().getString(R.string.pref_municipios_preferidos key)
100         , new HashSet<String>());
101     StringBuffer sb = new StringBuffer();
102     ArrayList<String> tempList = new ArrayList(temp);
103     for (int i = 0; i < tempList.size(); i++) {
104         int j = codMunicipio.indexOf(tempList.get(i));
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGeralActivity.java

```
104     sb.append(nomeMunicipio.get(j));
105     if (i < tempList.size() - 1) {
106         sb.append(", ");
107     }
108 }
109 sb.append(". ");
110 txtBackgroundMunicipios.setText("A busca será realizada apenas no(s) município(s): " + sb.toString());
111 }
112
113 @Override
114 public void informarTermoParaConsulta(int tipoBusca, String termo) {
115     barraDeProgresso.setVisibility(View.VISIBLE);
116     txtBackground.setVisibility(View.GONE);
117     txtBackgroundMunicipios.setVisibility(View.GONE);
118     meuRecyclerView.setVisibility(View.GONE);
119     imagemBackground.setVisibility(View.GONE);
120     carregadorDeDados = new CarregadorDeDados(this, tipoBusca, termo);
121     carregadorDeDados.execute();
122 }
123
124
125 @Override
126 public void onClick(View v) {
127     BuscaGeralDialog b = new BuscaGeralDialog();
128     b.setListaDeMunicipios(listaDeMunicipios);
129     b.show(this, getSupportFragmentManager(), "DialogBuscaGeral");
130 }
131
132 private class CarregadorDeDados extends AsyncTask<Void, Void, Void> {
133     private Context contexto;
134     private String termoBuscado;
135     private int tipoBusca;
136
137     public CarregadorDeDados(Context contexto, int tipoBusca, String termoBuscado) {
138         this.contexto = contexto;
139     }
140 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGeralActivity.java

```
139     this.termoBuscado = termoBuscado;
140     this.tipoBusca = tipoBusca;
141     Log.e("tipoDeBusca", "Tipo de busca= " + tipoBusca + " Termo: " + termoBuscado);
142 }
143
144 @Override
145 protected void doInBackground(Void... params) {
146     config = getSharedPreferences("favoritos", MODE_PRIVATE);
147     listaDeFavoritos = config.getStringSet("listaDeFavoritos", new HashSet<String>());
148     listaDeMunicipios = PreferenceManager.getDefaultSharedPreferences(contexto).getStringSet(R.
        string.pref_municipios_preferidos_key), new HashSet<String>());
149     estabelecimentos.clear();
150     switch (tipoBusca) {
151         case BuscaGeralDialog.SPINNER_CONVENIOS_ACEITOS:
152             AtendimentoDAO atendimentoDAOConv = AtendimentoDAO.getInstance(getApplicationContext());
153             estabelecimentos = atendimentoDAOConv.buscaStringEmConvenios(listaDeMunicipios, termoBuscado);
154             break;
155         case BuscaGeralDialog.SPINNER_FORMAS_DE_ATENDIMENTO:
156             AtendimentoDAO atendimentoDAOaten = AtendimentoDAO.getInstance(getApplicationContext());
157             estabelecimentos = atendimentoDAOaten.buscaStringEmAtendimentos(listaDeMunicipios, termoBuscado);
158             break;
159         case BuscaGeralDialog.SPINNER_MEDICOS:
160             termoBuscado = StringUtils.removeAcentos(termoBuscado);
161             termoBuscado = termoBuscado.trim();
162             termoBuscado = termoBuscado.toUpperCase();
163             ProfissionaisDAO profissionaisDAOmed = ProfissionaisDAO.getInstance(getApplicationContext());
164             estabelecimentos = profissionaisDAOmed.buscaStringEmNomeProfissionais(listaDeMunicipios,
                termoBuscado);
165             break;
166         case BuscaGeralDialog.SPINNER_ESPECIALIDADES:
167             ProfissionaisDAO profissionaisDAOesp = ProfissionaisDAO.getInstance(getApplicationContext());
168             estabelecimentos = profissionaisDAOesp.buscaStringEmNomeEspecialidade(listaDeMunicipios,
                termoBuscado);
169             break;
170         case BuscaGeralDialog.SPINNER_NOME_DO_ESTABELECIMENTO:
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_\dimon\cadesaude\BuscaGeralActivity.java

```
171 termoBuscado = StringUtils.removeAcentos(termoBuscado);
172 termoBuscado = termoBuscado.trim();
173 termoBuscado = termoBuscado.toUpperCase();
174 estabDAO = EstabelecimentoDAO.getInstance(getApplicationContext());
175 estabelecimentos = estabDAO.buscaStringEmEstabelecimentos(listaDeMunicipios, termoBuscado);
176 break;
177 case BuscaGeralDialog.SPINNER_SERVICOS_OFERECIDOS:
178     ServicoClassificacaoDAO servicoClassificacaoDAO = ServicoClassificacaoDAO.getInstance(
getApplicationContext());
179     estabelecimentos = servicoClassificacaoDAO.buscaStringEmServicos(listaDeMunicipios, termoBuscado);
180     break;
181     }
182 Collections.sort(estabelecimentos);
183 return null;
184 }
185
186 @Override
187 protected void onPostExecute(Void aVoid) {
188     super.onPostExecute(aVoid);
189     meuAdapter = new EstabelecimentoAdapter(contexto, estabelecimentos, listaDeFavoritos);
190     meuRecyclerView.setAdapter(meuAdapter);
191     if (estabelecimentos.size() == 0) {
192         txtBackground.setText(R.string.txt_erro_busca_geral);
193         txtBackground.setVisibility(View.VISIBLE);
194         imagemBackground.setVisibility(View.VISIBLE);
195         barraDeProgresso.setVisibility(View.GONE);
196         txtBackgroundMunicipios.setVisibility(View.GONE);
197         meuRecyclerView.setVisibility(View.GONE);
198     } else {
199         txtBackground.setVisibility(View.GONE);
200         imagemBackground.setVisibility(View.GONE);
201         barraDeProgresso.setVisibility(View.GONE);
202         meuRecyclerView.setVisibility(View.VISIBLE);
203     }
204     CoordinatorLayout cl = (CoordinatorLayout) findViewById(R.id.activity_busca_geral_layout);
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_\dimon\cadesaude\BuscaGeralActivity.java

```
205         Snackbar.make(c1, "Foram localizados " + estabelecimentos.size() + " estabelecimentos", Snackbar.  
        LENGTH_SHORT).show();  
206     }  
207     }  
208  
209     }  
210
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGeralDialog.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.app.Activity;
4 import android.app.AlertDialog;
5 import android.app.Dialog;
6 import android.app.DialogFragment;
7 import android.app.ProgressDialog;
8 import android.content.Context;
9 import android.content.DialogInterface;
10 import android.os.AsyncTask;
11 import android.os.Bundle;
12 import android.preference.PreferenceManager;
13 import android.text.Editable;
14 import android.text.TextWatcher;
15 import android.view.LayoutInflater;
16 import android.view.View;
17 import android.widget.AdapterView;
18 import android.widget.AdapterView.Adapter;
19 import android.widget.Button;
20 import android.widget.EditText;
21 import android.widget.Spinner;
22
23 import java.util.HashSet;
24 import java.util.Set;
25
26 /**
27  * Created by Jucemar on 13/04/2016.
28  */
29
30 public class BuscaGeralDialog extends DialogFragment implements DialogInterface.OnClickListener, TextWatcher,
31     AdapterView.OnItemClickListener {
32     private InterfaceBuscaGeralPorString comunicador;
33     private EditText campoConsulta;
34     private Button buttonConfirmar;
35     private Button buttonCancelar;
36     private ArrayAdapter<CharSequence> tiposDeBuscaAdapter;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGeralDialog.java

```
35 private Spinner comboBoxTipoBusca;
36 private Spinner comboBoxBusca;
37 private int tipoDeBusca;
38 private Set<String> listaDeMunicipios;
39 private String busca;
40 private BuscaGeralSpinnerAdapter valoresDeBuscaAdapter;
41 public static final int SPINNER_SELECCION_EUMA_OPCAO = 0;
42 public static final int SPINNER_CONVENIOS_ACEITOS = 1;
43 public static final int SPINNER_FORMAS_DE_ATENDIMENTO = 2;
44 public static final int SPINNER_MEDICOS = 3;
45 public static final int SPINNER_ESPECIALIDADES = 4;
46 public static final int SPINNER_NOME_DO_ESTABELECIMENTO = 5;
47 public static final int SPINNER_SERVICOS_OFERECIDOS = 6;
48
49 public void setListaDeMunicipios(Set<String> listaDeMunicipios) {
50     this.listaDeMunicipios = listaDeMunicipios;
51 }
52
53 public BuscaGeralDialog() {
54     this.tipoDeBusca = -1;
55 }
56
57 @Override
58 public void onAttach(Activity activity) {
59     super.onAttach(activity);
60     if (!(activity instanceof InterfaceBuscaGeralPorString)) {
61         throw new RuntimeException("A activity precisa implementar a interface InterfaceBuscaPorTermo");
62     }
63     comunicador = (InterfaceBuscaGeralPorString) activity;
64 }
65
66 @Override
67 public Dialog onCreateDialog(Bundle savedInstanceState) {
68     valoresDeBuscaAdapter = new BuscaGeralSpinnerAdapter(getActivity());
69 }
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGeralDialog.java

```
70 LayoutInflater l = (LayoutInflater) getActivity().getSystemService(Context.LAYOUT_INFLATER_SERVICE);
71 View v = l.inflate(R.layout.view_dialog_busca_geral, null);
72 AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
73 builder.setTitle(R.string.titulo_dialog_buscar_geral)
74     .setPositiveButton("Confirmar", this)
75     .setNegativeButton("Cancelar", this)
76     .setView(v);
77 campoConsulta = (EditText) v.findViewById(R.id.campo_busca_geral_dialog);
78 campoConsulta.addTextChangedListener(this);
79 tiposDeBuscaAdapter = new ArrayAdapter<CharSequence>(getActivity(), android.R.layout.simple_spinner_item);
80 comboBoxTipoBusca = (Spinner) v.findViewById(R.id.comboBox_tipo_busca_geral_dialog);
81 comboBoxBusca = (Spinner) v.findViewById(R.id.comboBox_busca_geral_dialog);
82 comboBoxTipoBusca.setOnItemSelectedListener(this);
83 comboBoxBusca.setOnItemSelectedListener(new EscutadorComboBoxBusca());
84 String[] itens = getResources().getStringArray(R.array.tipos_buscas);
85 for (String s : itens) {
86     tiposDeBuscaAdapter.add(s);
87 }
88 tiposDeBuscaAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
89 comboBoxTipoBusca.setAdapter(tiposDeBuscaAdapter);
90 return builder.create();
91 }
92
93 @Override
94 public void onStart() {
95     super.onStart();
96     AlertDialog dialog = (AlertDialog) getDialog();
97     dialog.setCanceledOnTouchOutside(false);
98     dialog.setCancelable(false);
99     this.buttonConfirmar = ((AlertDialog) dialog).getButton(DialogInterface.BUTTON_POSITIVE);
100     buttonConfirmar.setEnabled(false);
101     this.buttonCancelar = ((AlertDialog) dialog).getButton(DialogInterface.BUTTON_NEGATIVE);
102 }
103
104 @Override
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGeralDialog.java

```
105 public void onClick(DialogInterface dialog, int which) {
106     switch (which) {
107         case DialogInterface.BUTTON_NEGATIVE:
108             dialog.cancel();
109             break;
110         case DialogInterface.BUTTON_POSITIVE:
111             comunicador.informarTermoParaConsulta(tipoDeBusca, busca);
112             break;
113     }
114 }
115
116
117 @Override
118 public void beforeTextChanged(CharSequence s, int start, int count, int after) {
119 }
120
121 @Override
122 public void onTextChanged(CharSequence s, int start, int before, int count) {
123     if (campoConsulta.getVisibility() == View.VISIBLE) {
124         if (s.length() < 3) {
125             buttonConfirmar.setEnabled(false);
126         } else {
127             buttonConfirmar.setEnabled(true);
128         }
129         busca = s.toString();
130     }
131 }
132
133 @Override
134 public void afterTextChanged(Editable s) {
135 }
136
137 @Override
138 public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
139     tipoDeBusca = position;
140 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_\dimon\cadesaude\BuscaGeralDialog.java

```
140     switch (position) {
141         case
142             campoConsulta.setVisibility(View.VISIBLE);
143             comboBoxBusca.setVisibility(View.GONE);
144             break;
145         case SPINNER_NOME_DO_ESTABELECIMENTO:
146             campoConsulta.setVisibility(View.VISIBLE);
147             comboBoxBusca.setVisibility(View.GONE);
148             break;
149         case SPINNER_SELECCIONA_UMA_OPCAO:
150             campoConsulta.setText("");
151             campoConsulta.setVisibility(View.GONE);
152             comboBoxBusca.setVisibility(View.GONE);
153             buttonConfirmar.setEnabled(false);
154             break;
155         default:
156             campoConsulta.setVisibility(View.GONE);
157             valoresDeBuscaAdapter.clear();
158             CarregadorDeDados carregadorDeDados = new CarregadorDeDados(getActivity(), position);
159             carregadorDeDados.execute();
160         }
161
162     }
163
164     @Override
165     public void onNothingSelected(AdapterView<?> parent) {
166         //Não há necessidade de implementar, o android chama esse método somente se a dialog dismiss ou se o adapter
167         tem tamanho==0
168     }
169
170     private class CarregadorDeDados extends AsyncTask<Void, Void, Void> {
171         private int tipoBusca;
172         private CharSequence[] dadosSpinner02;
173         private ProgressDialog pd;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGeralDialog.java

```
174 private Context contexto;
175
176 public CarregadorDeDados(Context contexto, int tipoBusca) {
177     this.tipoBusca = tipoBusca;
178     this.contexto = contexto;
179 }
180
181 @Override
182 protected void onPreExecute() {
183     super.onPreExecute();
184     pd = ProgressDialog.show(contexto, "Carregando", "Aguarde", true, true);
185     pd.setOnCancelListener(new DialogInterface.OnCancelListener() {
186
187         @Override
188         public void onCancel(DialogInterface dialog) {
189             try {
190                 this.finalize();
191             } catch (Throwable throwable) {
192                 throwable.printStackTrace();
193             }
194         });
195     }
196
197     @Override
198     protected void doInBackground(Void... params) {
199         ListaDeMunicipios = PreferenceManager.getDefaultSharedPreferences(getActivity())
200             .getStringSet(getString(R.string.pref_municipios_preferidos_key), new HashSet<String>());
201         switch (tipoDeBusca) {
202             case SPINNER_CONVENIOS_ACEITOS:
203                 AtendimentoDAO atendimentoDAOConv = AtendimentoDAO
204                     .getInstance(contexto.getApplicationContext());
205                 dadosSpinner02 = atendimentoDAOConv.listarTiposConvenios(listaDeMunicipios);
206                 break;
207             case SPINNER ESPECIALIDADES:
208                 ---

```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGeralDialog.java

```
209     ProfissionaisDAO profissionaisDAO = ProfissionaisDAO
210         .getInstancia(contexto.getApplicationContext());
211     dadosSpinner02 = profissionaisDAO.listarTiposProfissionais(listaDeMunicipios);
212     break;
213     case SPINNER_FORMAS_DE_ATENDIMENTO:
214         AtendimentoDAO atendimentoDAOAtend = AtendimentoDAO
215             .getInstancia(contexto.getApplicationContext());
216         dadosSpinner02 = atendimentoDAOAtend.listarTiposAtendimento(listaDeMunicipios);
217         break;
218     case SPINNER_SERVICOS_OFERECIDOS:
219         ServicoClassificacaoDAO servicoClassificacaoDAO = ServicoClassificacaoDAO
220             .getInstancia(contexto.getApplicationContext());
221         dadosSpinner02 = servicoClassificacaoDAO.listarTiposServicos(listaDeMunicipios);
222         break;
223     }
224     return null;
225 }
226
227 @Override
228 protected void onPostExecute(Void aVoid) {
229     super.onPostExecute(aVoid);
230     if (pd.isShowing())
231         pd.dismiss();
232     valoresDeBuscaAdapter.addAll(dadosSpinner02);
233     valoresDeBuscaAdapter.setDropDownViewResource(R.layout.item_spinner);
234     comboBoxBusca.setAdapter(valoresDeBuscaAdapter);
235     comboBoxBusca.setVisibility(View.VISIBLE);
236 }
237
238 }
239
240 private class EscutadorComboBoxCampoBusca implements AdapterView.OnItemClickListener {
241     @Override
242     public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
243         ...

```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGeralDialog.java

```
244     if (position > 0) {
245         buttonConfirmar.setEnabled(true);
246     } else {
247         buttonConfirmar.setEnabled(false);
248     }
249     busca = (String) parent.getItemAtPosition(position);
250 }
251
252 @Override
253 public void onNothingSelected(AdapterView<?> parent) {
254 }
255 }
256
257 }
258
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juccemar_dimon\cadesaude\BuscaGeralSpinnerAdapter.java

```
1 package br.com.juccemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.ArrayAdapter;
8 import android.widget.TextView;
9
10 import java.util.ArrayList;
11
12 /**
13  * Created by Juccemar on 07/05/2016.
14  */
15 public class BuscaGeralSpinnerAdapter extends ArrayAdapter<CharSequence> {
16
17     private Context contexto;
18     private ArrayList<String> spinnerValues;
19
20     public BuscaGeralSpinnerAdapter(Context context, ArrayList<String> spinnerValues) {
21         super(context, R.layout.item_spinner);
22         this.contexto = context;
23         this.spinnerValues = spinnerValues;
24     }
25
26     public BuscaGeralSpinnerAdapter(Context context) {
27         super(context, R.layout.view_dialog_busca_geral);
28         this.contexto = context;
29     }
30
31     @Override
32     public View getView(int pos, View convertView, ViewGroup parent) {
33         return getCustomView(pos, convertView, parent);
34     }
35 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSPesquisaSpinnerAdapter.java

```
36 package br.com.jucemar_dimon.cadesaude;
37 /**
38  * Created by mySpinner on 06/11/16.
39  */
40 TextView main_text = (TextView) mySpinner.findViewById(R.id.item_spinner_busca_geral);
41 main_text.setText(getItem(position));
42 import android.os.Bundle;
43 import android.app.Activity;
44 import android.app.AlertDialog;
45 import android.app.SearchManager;
46 import android.content.Context;
47 import android.content.Intent;
48 import android.content.IntentSender;
49 import android.content.SharedPreferences;
50 import android.content.pm.PackageManager;
51 import android.graphics.Color;
52 import android.location.Location;
53 import android.os.AsyncTask;
54 import android.os.Bundle;
55 import android.preference.PreferenceManager;
56 import android.support.design.widget.CoordinatorLayout;
57 import android.support.design.widget.Snackbar;
58 import android.support.v4.app.ActivityCompat;
59 import android.support.v4.view.MenuItemCompat;
60 import android.support.v7.app.AppCompatActivity;
61 import android.support.v7.widget.DefaultItemAnimator;
62 import android.support.v7.widget.LinearLayoutManager;
63 import android.support.v7.widget.RecyclerView;
64 import android.support.v7.widget.SearchView;
65 import android.support.v7.widget.Toolbar;
66 import android.util.Log;
67 import android.view.Menu;
68 import android.view.MenuInflater;
69 import android.view.MenuItem;
70 import android.view.View;
71 ... ..
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
36 import android.widget.ProgressBar;
37 import android.widget.TextView;
38
39 import com.github.clans.fab.FloatingActionButton;
40 import com.github.clans.fab.FloatingActionMenu;
41 import com.google.android.gms.common.ConnectionResult;
42 import com.google.android.gms.common.GoogleApiAvailability;
43 import com.google.android.gms.common.api.GoogleApiClient;
44 import com.google.android.gms.common.api.GoogleApiClient.ConnectionCallbacks;
45 import com.google.android.gms.common.api.PendingResult;
46 import com.google.android.gms.common.api.ResultCallback;
47 import com.google.android.gms.common.api.Status;
48 import com.google.android.gms.location.LocationListener;
49 import com.google.android.gms.location.LocationRequest;
50 import com.google.android.gms.location.LocationServices;
51 import com.google.android.gms.location.LocationSettingsRequest;
52 import com.google.android.gms.location.LocationSettingsResult;
53 import com.google.android.gms.location.LocationSettingsStates;
54 import com.google.android.gms.location.LocationSettingsStatusCodes;
55
56 import java.util.ArrayList;
57 import java.util.Arrays;
58 import java.util.HashSet;
59 import java.util.Set;
60
61 public class BuscaGPSActivity extends AppCompatActivity implements
62     CaixaDialogoGoogleAPI.ErroGooglePlayService,
63     ConnectionCallbacks,
64     GoogleApiClient.OnConnectionFailedListener,
65     LocationListener, ResultCallback<LocationSettingsResult>,
66     View.OnClickListener,
67     InterfaceFiltroDeEstabelecimentos {
68
69     public static final int REQUEST_CODE_GOOGLE_PLAY_SERVICES = 1982;
70 }
```

```

71 private static final int REQUEST_CODE_LOCATION = 2;
72 private Toolbar toolbar;
73 private String distanciaMaxima;
74 private Status status;
75 private String dadosDeBuscaNaTela;
76 private Location localizacao;
77 private Double[] latLon;
78 private LocationRequest request;
79 private GoogleApiClient apiClient;
80 private EstabelecimentoDAO estabDAO;
81 private ArrayList<Estabelecimento> estabelecimentos;
82 private ArrayList<Estabelecimento> estabelecimentosFiltrados;
83 private RecyclerView meuRecyclerView;
84 private EstabelecimentoAdapter meuAdapter;
85 private RecyclerView.LayoutManager meuLayoutManager;
86 private Set<String> listaDeFavoritos;
87 private Set<String> listaDeUnicipios;
88 private SharedPreferences config;
89 private FloatingActionButton fabFiltroTipoEstab;
90 private FloatingActionButton fabFiltroTipoAtend;
91 private FloatingActionButton fabFiltroTipoConv;
92 private FloatingActionButton fabFiltroTipoServ;
93 private FloatingActionButton fabFiltroTipoProf;
94 private FloatingActionButton fabFiltroMenu;
95 private FloatingActionButton fabFiltroLimparTodos;
96 private ProgressBar barraDeProgresso;
97 private TextView txtErroGps;
98 private CarregadorDeDadosBGPSA carregadorDeDados;
99 private int posicaoNaLista;
100 private MenuItem botaoGps;
101 private MenuItem botaoBusca;
102 private ArrayList<Estabelecimento> bkpEstabelecimentos;
103 private ArrayList<String> tiposEstabelecimentosSelecionados;
104 private ArrayList<String> tiposAtendimentosSelecionados;

```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
106 private ArrayList<String> tiposConveniosSelecionados;
107 private ArrayList<String> tiposProfissionaisSelecionados;
108 private ArrayList<String> tiposServicosSelecionados;
109 private ArrayList<Integer> filtrosAtivos;
110 private CharSequence[] labelsTiposProfissionais;
111 private CharSequence[] labelsTiposEstabelecimentos;
112 private CharSequence[] labelsTiposAtendimentos;
113 private CharSequence[] labelsTiposConvenios;
114 private CharSequence[] labelsTiposServicos;
115 private boolean coordenadasAlteradas;
116 private CoordinatorLayout layout;
117
118 //Método que verifica se há na intent uma string armazenada cuja chave é a action de serchview
119 private void executarBuscaPorPalavra(Intent intent) {
120     if (Intent.ACTION_SEARCH.equals(intent.getAction())) {
121         dadosDeBuscaNaTela = intent.getStringExtra(SearchManager.QUERY);
122     }
123 }
124
125 @Override
126 protected void onCreate(Bundle savedInstanceState) {
127     super.onCreate(savedInstanceState);
128     setContentView(R.layout.activity_busca_gps);
129
130     //instancia os objetos do layout
131     barraDeProgresso = (ProgressBar) findViewById(R.id.barra_progresso_gps_activity);
132     toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_busca_gps);
133     txtErroGps = (TextView) findViewById(R.id.label_erro_gps);
134     imgErroGps = (ImageView) findViewById(R.id.background_erro_gps);
135     fabFiltroTipoEstab = (FloatingActionButton) findViewById(R.id.fab_filtro_tipo_estab);
136     fabFiltroTipoAtend = (FloatingActionButton) findViewById(R.id.fab_filtro_tipo_atend);
137     fabFiltroTipoConv = (FloatingActionButton) findViewById(R.id.fab_filtro_tipo_conv);
138     fabFiltroTipoServ = (FloatingActionButton) findViewById(R.id.fab_filtro_tipo_serv);
139     fabFiltroTipoProf = (FloatingActionButton) findViewById(R.id.fab_filtro_tipo_prof);
140     fabFiltroMenu = (FloatingActionButton) findViewById(R.id.fab_filtro_menu);
141 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
141 fabFiltroLimparTodos = (FloatingActionButton) findViewById(R.id.fab_filtro_limpar_todos);
142
143 //configura objetos do layout
144 barraDeProgresso.setVisibility(View.VISIBLE);
145 toolbar.setTitle("Buscar com GPS");
146
147 setSupportActionBar(toolbar);
148
149 getSupportActionBar().setDisplayHomeAsUpEnabled(true);
150 txtErroGps.setVisibility(View.GONE);
151 imgErroGps.setVisibility(View.GONE);
152 fabFiltroTipoConv.setOnClickListener(this);
153 fabFiltroTipoServ.setOnClickListener(this);
154 fabFiltroTipoProf.setOnClickListener(this);
155 fabFiltroTipoAtend.setOnClickListener(this);
156 fabFiltroTipoEstab.setOnClickListener(this);
157 fabFiltroLimparTodos.setOnClickListener(this);
158 fabFiltroMenu.setOnClickListener(true);
159 layout = (CoordinatorLayout) findViewById(R.id.activity_busca_ons_layout);
160
161 //instancia outros objetos
162 coordenadasAlteradas = false;
163 filtrosAtivos = new ArrayList<Integer>();
164 latLon = new Double[2];
165 dadosDeBuscaNaTela = "";
166 posicaoNaLista = -1;
167 this.estabelecimentosFiltrados = new ArrayList<Estabelecimento>();
168
169 //restaura os dados, caso hajam, dos filtros que foram salvos anteriormente caso tenha havido alguma ação no
170 dispositivo que tenha posto outra activity sobre a atual
171 if (savedInstanceState != null) {
172     posicaoNaLista = savedInstanceState.getInt("posicaoNaLista");
173     if (savedInstanceState.getParcelableArrayList("estabelecimentos") != null) {
174         estabelecimentos = savedInstanceState.getParcelableArrayList("estabelecimentos");
175     }
176 }
177 if (savedInstanceState.getStringArrayList("tiposEstabelecimentosSelecionados") != null) {
178     tiposEstabelecimentosSelecionados = savedInstanceState.getStringArrayList("

```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
174 tiposEstabelecimentosSelecionados");
175     } else {
176         this.tiposEstabelecimentosSelecionados = new ArrayList<String>();
177     }
178     if (savedInstanceState.getIntegerArrayList("filtrosAtivos") != null) {
179         this.filtrosAtivos = savedInstanceState.getIntegerArrayList("filtrosAtivos");
180     } else {
181         this.filtrosAtivos = new ArrayList<Integer>();
182     }
183     if (savedInstanceState.getStringArrayList("tiposAtendimentosSelecionados") != null) {
184         tiposAtendimentosSelecionados = savedInstanceState.getStringArrayList("tiposAtendimentosSelecionados")
        ;
185     } else {
186         this.tiposAtendimentosSelecionados = new ArrayList<String>();
187     }
188     if (savedInstanceState.getStringArrayList("tiposConveniosSelecionados") != null) {
189         tiposConveniosSelecionados = savedInstanceState.getStringArrayList("tiposConveniosSelecionados");
190     } else {
191         this.tiposConveniosSelecionados = new ArrayList<String>();
192     }
193     if (savedInstanceState.getStringArrayList("tiposProfissionaisSelecionados") != null) {
194         tiposProfissionaisSelecionados = savedInstanceState.getStringArrayList("tiposProfissionaisSelecionados
        ");
195     } else {
196         this.tiposProfissionaisSelecionados = new ArrayList<String>();
197     }
198     if (savedInstanceState.getStringArrayList("tiposServicosSelecionados") != null) {
199         tiposServicosSelecionados = savedInstanceState.getStringArrayList("tiposServicosSelecionados");
200     } else {
201         this.tiposServicosSelecionados = new ArrayList<String>();
202     }
203     } else {
204         this.tiposServicosSelecionados = new ArrayList<String>();
205         this.tiposProfissionaisSelecionados = new ArrayList<String>();
206         this.tiposConveniosSelecionados = new ArrayList<String>();
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
207     this.tiposAtendimentosSelecionados = new ArrayList<String>();
208     this.tiposEstabelecimentosSelecionados = new ArrayList<String>();
209 }
210
211 //metodo chamado para para realizar verificações para saber se o dispositivo possui o google play services
    instalado
212     verificarGooglePlayService();
213
214 //Instancia o objeto da api do google e apartir de dai realiza checagens quanto ao google play services
215     apiClient = new GoogleApiClient.Builder(this)
216         .addConnectionCallbacks(this)
217         .addOnConnectionFailedListener(this)
218         .addApi(LocationServices.API)
219         .build();
220 }
221
222     private void mostrarQtdeResultados() {
223         Snackbar.make(layout, "Foram localizados " + meuAdapter.getItemCount() + " estabelecimentos", Snackbar.
LENGTH_SHORT).show();
224     }
225
226     @Override
227     public boolean onOptionsItemSelected(MenuItem item) {
228         int id = item.getItemId();
229         switch (id) {
230             case android.R.id.home:
231                 onBackPressed();
232                 break;
233             case R.id.action_gps: fabFiltroMenu.close(false);
234                 coordenadasAlteradas = true;
235                 meuRecyclerView.setVisibility(View.GONE);
236                 fabFiltroMenu.setVisibility(View.GONE);
237                 barraDeProgresso.setVisibility(View.VISIBLE);
238                 botaoBusca.setVisible(false);
239 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
240     botaoGps.setVisibility(false);
241     adquirirLocalizacao();
242 }
243 return super.onOptionsItemSelected(item);
244 }
245
246
247 //metodo que cria e mostra os dados na lista caso todas as etapas de verificação do correto funcionamento da api
de localização do google tenham sido aprovadas
248 private void abrirRecyclerView() {
249
250     //barraDeProgresso.setVisibility(View.VISIBLE);
251     fabFiltroMenu.setVisibility(View.GONE);
252     meuRecyclerView = (RecyclerView) findViewById(R.id.lista_estabelecimentos);
253     meuRecyclerView.setVisibility(View.GONE);    meuLayoutManager = new
254     LinearLayoutManager(this);
255     meuRecyclerView.setLayoutManager(meuLayoutManager);
256     meuRecyclerView.setItemAnimator(new    DefaultItemAnimator());
257     meuRecyclerView.setHasFixedSize(true);
258     meuRecyclerView.addOnScrollListener(new RecyclerView.OnScrollListener() {
259
260         @Override
261         public void onScrollStateChanged(RecyclerView recyclerView, int newState) {
262             super.onScrollStateChanged(recyclerView, newState);
263             Log.e("Scrool", String.valueOf(newState));
264         }
265     })
266
267     @Override
268     public void onScrolled(RecyclerView recyclerView, int dx, int dy) {
269         super.onScrolled(recyclerView, dx, dy);
270         if (dy > 0) {
271             fabFiltroMenu.hideMenuButton(true);
272         } else {
273             fabFiltroMenu.showMenuButton(true);
274         }
275     }
276 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_\dimon\cadesaude\BuscaGPSActivity.java

```
274     }
275     });
276     executarBuscaPorPalavra(getIntent());
277     carregarDados();
278     fabFiltroMenu.showMenuButton(true);
279 }
280
281 private void carregarDados() {
282     carregadorDeDados = new CarregadorDeDadosBGPSA(this);
283     carregadorDeDados.execute();
284 }
285
286 @Override
287 protected void onSaveInstanceState(Bundle outState) {
288     super.onSaveInstanceState(outState);
289     outState.putParcelableArrayList("estabelecimentos", estabelecimentos);
290     outState.putStringArrayList("tiposEstabelecimentosSelecionados", tiposEstabelecimentosSelecionados);
291     outState.putStringArrayList("tiposAtendimentosSelecionados", tiposAtendimentosSelecionados);
292     outState.putStringArrayList("tiposConveniosSelecionados", tiposConveniosSelecionados);
293     outState.putStringArrayList("tiposProfissionaisSelecionados", tiposProfissionaisSelecionados);
294     outState.putStringArrayList("tiposServicosSelecionados", tiposServicosSelecionados);
295     outState.putIntArrayList("filtrosAtivos", filtrosAtivos);
296     outState.putInt("posicaoNaLista", posicaoNaLista);
297 }
298
299 private void verificarGooglePlayService() {
300     GoogleApiAvailability api = GoogleApiAvailability.getInstance();
301     int code = api.isGooglePlayServicesAvailable(this);
302     if (code == ConnectionResult.SUCCESS) {
303         //da sequencia nos procedimento normalmente
304     } else {
305         if (code == ConnectionResult.SERVICE_MISSING ||
306             code == ConnectionResult.SERVICE_VERSION_UPDATE_REQUIRED ||
307             code == ConnectionResult.SERVICE_DISABLED) {
308             Dialog dialog = api.getErrorDialog(this, code, REQUEST_CODE_GOOGLE_PLAY_SERVICES);
309             ...

```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_\dimon\cadesaude\BuscaGPSActivity.java

```
309         dialog.show();
310     } else {
311         CaixaDeDialogoGoogIeAPI cxDialog = new CaixaDeDialogoGoogIeAPI();
312         cxDialog.setErroGoogIePlayService(this);
313         cxDialog.show(getFragmentManager(), "DialogErro");
314     }
315 }
316 }
317
318 @Override
319 protected void onNewIntent(Intent intent) {
320     super.onNewIntent(intent);
321     executarBuscaPorPalavra(intent);
322     carregarDados();
323 }
324
325 @Override
326 public boolean onCreateOptionsMenu(Menu menu) { MenuInflater
327     menuInflater = getMenuInflater();
328     menuInflater.inflate(R.menu.consulta_gps, menu);
329     MenuItem searchItem = menu.findItem(R.id.action_search);
330     botaoBusca = searchItem;
331     botaoGps = menu.findItem(R.id.action_gps);
332     SearchManager searchManager = (SearchManager) BuscaGPSActivity.this.getSystemService(Context.SEARCH_SERVICE);
333     SearchView searchView = null;
334     if (searchItem != null) {
335         searchView = (SearchView) searchItem.getActionView();
336     }
337     if (searchView != null) {
338         searchView.setSearchableInfo(searchManager.getSearchableInfo(this.getComponentName()));
339     }
340     MenuItemCompat.setOnActionExpandListener expandListener = new MenuItemCompat.OnActionExpandListener() {
341         @Override
342         public boolean onMenuItemActionCollapse(MenuItem item) {
343             botaoGps.setVisible(true);
344         }
345     }
346 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
344     dadosDeBuscaNaTela = "";  
345     fabFiltroMenu.close(true);  
346     carregarDados();  
347     return true;  
348 }  
349  
350 @Override  
351 public boolean onOptionsItemSelected(MenuItem item) {  
352     botaoGps.setVisible(false);  
353     fabFiltroMenu.close(true);  
354     return true;  
355 }  
356  
357 MenuItemCompat.setOnActionExpandListener(searchItem, expandListener);  
358 return super.onCreateOptionsMenu(menu);  
359 }  
360  
361  
362 @Override  
363 protected void onStart() {  
364     super.onStart();  
365     apiClient.connect();  
366 }  
367  
368 @Override  
369 public void onBackPressed() {  
370     barraDeProgresso.setVisibility(View.GONE);  
371     if (fabFiltroMenu.isOpened()) {  
372         fabFiltroMenu.toggle(true);  
373     } else {  
374         if (carregadorDeDados != null) {  
375             carregadorDeDados.cancel(true);  
376         }  
377         super.onBackPressed();  
378     }  
379 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
379     }
380
381     @Override
382     protected void onStop() {
383         super.onStop();
384         apiClient.disconnect();
385     }
386
387     @Override
388     protected void onPause() {
389         super.onPause();
390         if (meuRecyclerView != null) {
391             LinearLayoutManager lm = (LinearLayoutManager) meuRecyclerView.getLayoutManager();
392             posicaoNaLista = lm.findFirstVisibleItemPosition();
393         }
394     }
395
396     @Override
397     public void onConnected(Bundle bundle) { request =
398         LocationRequest.create();
399         request.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
400         request.setInterval(500);
401         LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder().addLocationRequest(request);
402         builder.setAlwaysShow(true);
403         PendingResult<LocationSettingsResult> result = LocationServices.SettingsApi.checkLocationSettings(apiClient,
404             builder.build());
405         result.setResultCallback(this);
406     }
407
408     public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
409         if (requestCode == REQUEST_CODE_LOCATION) {
410             if (grantResults.length == 1 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
411                 adquirirLocalizacao();
412             } else {
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_\dimon\cadesaude\BuscaGPSActivity.java

```
413 barraDeProgresso.setAlpha(0f);
414 imgErroGps.setVisibility(View.VISIBLE);
415 txtErroGps.setVisibility(View.VISIBLE);
416 Snackbar.make(fabFiltroMenu, R.string.tentar_novamente_label, Snackbar.LENGTH_INDEFINITE)
417     .setAction(R.string.tentar_novamente_action, new View.OnClickListener() {
418         @Override
419         public void onClick(View v) {
420             adquirirLocalizacao();
421         }
422     })
423     .setActionTextColor(Color.YELLOW)
424     .show();
425     }
426 }
427 }
428
429 private void adquirirLocalizacao() {
430     if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.
431         PERMISSION_GRANTED) {
432         if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.ACCESS_FINE_LOCATION)) {
433             ActivityCompat.requestPermissions(this, new String[] {Manifest.permission.ACCESS_FINE_LOCATION,
434                 REQUEST_CODE_LOCATION});
435         } else {
436             LocationServices.FusedLocationApi.requestLocationUpdates(apiClient, request, this);
437         }
438     }
439
440     @Override
441     public void onConnectionSuspended(int i) {
442         //Não implementado
443     }
444
445     @Override
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
446 public void onConnectionFailed(ConnectionResult connectionResult) {
447     //Não implementado
448 }
449
450 @Override
451 public void onLocationChanged(Location location) {
452     this.localizacao = location;
453     latLon[0] = location.getLatitude();
454     latLon[1] = location.getLongitude();
455     Log.e("Coordenadas", latLon[0] + " " + latLon[1]);
456     LocationServices.FusedLocationApi.removeLocationUpdates(apiClient, this);
457     abrirRecyclerView();
458 }
459
460 @Override
461 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
462     final LocationSettingsStates states;
463     if (data != null) {
464         states = LocationSettingsStates.fromIntent(data);
465     }
466     switch (requestCode) {
467         case REQUEST_CODE_GOOGLE_PLAY_SERVICES:
468             switch (resultCode) {
469                 case Activity.RESULT_OK:
470                     barraDeProgresso.setAlpha(1f);
471                     barraDeProgresso.setVisibility(View.VISIBLE);
472                     imgErroGps.setVisibility(View.GONE);
473                     txtErroGps.setVisibility(View.GONE);
474                     adquirirLocalizacao();
475                     break;
476                 case Activity.RESULT_CANCELED:
477                     barraDeProgresso.setAlpha(0f);
478                     imgErroGps.setVisibility(View.VISIBLE);
479                     txtErroGps.setVisibility(View.VISIBLE);
480                     Snackbar.make(fabFiltroMenu, R.string.tentar_novamente_label, Snackbar.LENGTH_INDEFINITE)
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
481 .setAction(R.string.tentar_novamente_action, new View.OnClickListener() {
482     @Override
483     public void onClick(View v) {
484         try {
485             status.startResolutionForResult(BuscaGPSActivity.this,
486                 REQUEST_CODE_GOOGLE_PLAY_SERVICES);
487         } catch (IntentSender.SendIntentException e) {
488             e.printStackTrace();
489         }
490     })
491     .setActionTextColor(Color.YELLOW)
492     .show();
493 }
494 break;
495 }
496 }
497
498 @Override
499 public void onResult(LocationSettingsResult locationSettingsResult) {
500     status = locationSettingsResult.getStatus();
501     switch (status.getStatusCode()) {
502         case LocationSettingsStatusCodes.SUCCESS:
503             adquirirLocalizacao();
504             break;
505         case LocationSettingsStatusCodes.RESOLUTION_REQUIRED:
506             try {
507                 status.startResolutionForResult(BuscaGPSActivity.this, REQUEST_CODE_GOOGLE_PLAY_SERVICES);
508             } catch (IntentSender.SendIntentException e) {
509                 Snackbar.make(findViewById(android.R.id.content), "Algun motivo desconhecido impediu a aquisição
510 de sua localização", Snackbar.LENGTH_LONG).setActionTextColor(Color.RED).show();
511             }
512             break;
513         case
514             LocationSettingsStatusCodes.SETTINGS_CHANGE_UNAVAILABLE:
515             Snackbar.make(findViewById(android.R.id.content), "Algun motivo desconhecido impediu a aquisição de
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
513 sua localização", Snackbar.LENGTH_LONG).setActionTextColor(Color.RED).show();
514     break;
515 }
516 }
517
518 @Override
519 public void onClick(View v) {
520     definirLabelsParaFiltros();
521     int id = v.getId();
522     switch (id) {
523         case R.id.fab_filtro_limpar_todos:
524             fabFiltroMenu.toggle(true);
525             limparTodosFiltros();
526             break;
527         case R.id.fab_filtro_tipo_atend:
528             fabFiltroMenu.toggle(true);
529             FiltroGeralDialog dialogFiltroTipoAtendimento = new FiltroGeralDialog();
530             dialogFiltroTipoAtendimento.setArgumentos(tiposAtendimentosSelecionados,
531                 InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ATENDIMENTO, labelsTiposAtendimentos);
532             dialogFiltroTipoAtendimento.show(getFragmentManager(), "DialogFiltroTipoAtendimento");
533             break;
534         case R.id.fab_filtro_tipo_conven:
535             fabFiltroMenu.toggle(true);
536             FiltroGeralDialog dialogFiltroTipoConvenio = new FiltroGeralDialog();
537             dialogFiltroTipoConvenio.setArgumentos(tiposConveniosSelecionados, InterfaceFiltroDeEstabelecimentos.
538                 FILTRO_TIPO_CONVENIO, labelsTiposConvenios);
539             dialogFiltroTipoConvenio.show(getFragmentManager(), "DialogFiltroTipoConvenio");
540             break;
541         case R.id.fab_filtro_tipo_prof:
542             fabFiltroMenu.toggle(true);
543             FiltroGeralDialog dialogFiltroTipoProfissional = new FiltroGeralDialog();
544             dialogFiltroTipoProfissional.setArgumentos(tiposProfissionaisSelecionados,
545                 InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_PROFISSIONAL, labelsTiposProfissionais);
546             dialogFiltroTipoProfissional.show(getFragmentManager(), "DialogFiltroTipoProfissional");
547             break;
548     }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
545         case R.id.fab_filtro_tipo_serv:
546             fabFiltroMenu.toggle(true);
547             FiltroGeralDialog dialogFiltroTipoServico = new FiltroGeralDialog();
548             dialogFiltroTipoServico.setArgumentos(tiposServicosSelecionados, InterfaceFiltroDeEstabelecimentos.
549                 FILTRO_TIPO_SERVICO, labelsTiposServicos);
550             dialogFiltroTipoServico.show(getFragmentManager(), "DialogFiltroTipoServico");
551             break;
552         case R.id.fab_filtro_tipo_estab:
553             fabFiltroMenu.toggle(true);
554             FiltroGeralDialog dialogFiltroTipoEstabelecimento = new FiltroGeralDialog();
555             dialogFiltroTipoEstabelecimento.setArgumentos(tiposEstabelecimentosSelecionados,
556                 InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO, labelsTiposEstabelecimentos);
557             dialogFiltroTipoEstabelecimento.show(getFragmentManager(), "DialogFiltroTipoEstabelecimento");
558             break;
559         }
560     }
561     @Override
562     public void filtrarPorTipoDeEstabelecimento(ArrayList<String> tiposEstabelecimentosSelecionados) {
563         this.tiposEstabelecimentosSelecionados = tiposEstabelecimentosSelecionados;
564         ArrayList<Estabelecimento> estabelecimentosTemp = new ArrayList<Estabelecimento>();
565         ArrayList<Estabelecimento> estabelecimentosFiltradosTemp = new ArrayList<Estabelecimento>();
566         if (filtrosAtivos.size() > 0) {
567             estabelecimentosTemp = this.estabelecimentosFiltrados;
568         } else {
569             estabelecimentosTemp = this.estabelecimentos;
570         }
571         if (this.tiposEstabelecimentosSelecionados.size() > 0) {
572             fabFiltroMenu.setMenuItemColorPressedResId(R.color.accent);
573             fabFiltroMenu.setMenuItemColorNormalResId(R.color.primary);
574             fabFiltroTipoEstab.setColorNormalResId(R.color.primary);
575             fabFiltroTipoEstab.setColorPressedResId(R.color.accent);
576         }
577         for (Estabelecimento e : estabelecimentosTemp) {
578             if (this.tiposEstabelecimentosSelecionados.contains(e.getTipoEstabelecimento())) {
```


File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
578         estabelecimentosFiltradosTemp.add(e);
579     }
580 }
581 meuAdapter = new EstabelecimentoAdapter(this, estabelecimentosFiltradosTemp, listaDeFavoritos,
    distancia\laxima);
582 meuRecyclerView.invalidate();
583 meuRecyclerView.setAdapter(meuAdapter);
584 if (!filtrosAtivos.contains(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO)) {
585     filtrosAtivos.add(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO);
586     botaoBusca.setVisible(false);
587 }
588 this.estabelecimentosFiltrados = estabelecimentosFiltradosTemp;
589 definirLabelsParaFiltros();
590 mostraQtdeResultados();
591 }
592 }
593
594 private void restaurarFiltros(int filtro) {
595     switch (filtro) {
596         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ATENDIMENTO:
597             filtrarPorTipoDeEstabelecimento(this.tiposEstabelecimentosSelecionados);
598             break;
599         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_SERVICO:
600             filtrarPorTipoDeServico(this.tiposServicosSelecionados);
601             break;
602         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO:
603             filtrarPorTipoDeEstabelecimento(this.tiposEstabelecimentosSelecionados);
604             break;
605         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_PROFISSIONAL:
606             filtrarPorTipoDeProfissional(this.tiposProfissionaisSelecionados);
607             break;
608         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_CONVENIO:
609             filtrarPorTipoDeConvenio(this.tiposConveniosSelecionados);
610             break;
611     }
612 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
612     }
613
614     @Override
615     public void filtrarPorTipoDeAtendimento(ArrayList<String> tiposAtendimentosSelecionados) {
616         ArrayList<Estabelecimento> estabelecimentosTemp = new ArrayList<Estabelecimento>();
617         ArrayList<Estabelecimento> estabelecimentosFiltradosTemp = new ArrayList<Estabelecimento>();
618         int compatibilidade = 0;
619         this.tiposAtendimentosSelecionados = tiposAtendimentosSelecionados;
620         if (estabelecimentosFiltrados.size() > 0) {
621             estabelecimentosTemp = this.estabelecimentosFiltrados;
622         } else {
623             estabelecimentosTemp = this.estabelecimentos;
624         }
625         if (this.tiposAtendimentosSelecionados.size() > 0) {
626             for (Estabelecimento e : estabelecimentosTemp) {
627                 for (Atendimento a : e.getAtendimento()) {
628                     if (this.tiposAtendimentosSelecionados.contains(a.getTipoAtendimento())) {
629                         compatibilidade++;
630                     }
631                 }
632                 if (compatibilidade > 0) {
633                     fabFiltroMenu.setMenuButtonColorPressedResId(R.color.accent);
634                     fabFiltroMenu.setMenuButtonColorNormalResId(R.color.primary);
635                     fabFiltroTipoAtend.setColorNormalResId(R.color.primary);
636                     fabFiltroTipoAtend.setColorPressedResId(R.color.accent);
637                     estabelecimentosFiltradosTemp.add(e);
638                     compatibilidade = 0;
639                 }
640             }
641             meuAdapter = new EstabelecimentoAdapter(this, estabelecimentosFiltradosTemp, listaDeFavoritos,
642                 distanciaMaxima);
643             meuRecyclerView.invalidate();
644             meuRecyclerView.setAdapter(meuAdapter);
645             if (!filtrosAtivos.contains(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ATENDIMENTO)) {
646                 filtrosAtivos.add(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ATENDIMENTO);
647             }
648         }
649     }
650 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
646         botaoBusca.setVisible(false);
647     }
648     this.estabelecimentosFiltrados = estabelecimentosFiltradosTemp;
649     definirLabelsParaFiltros();
650     mostraQtdeResultados();
651 }
652 }
653
654 private void definirLabelsParaFiltros() {
655     ArrayList<Estabelecimento> estabelecimentosFiltradosTemp = new ArrayList<Estabelecimento>();
656     if (estabelecimentosFiltrados.size() > 0) {
657         estabelecimentosFiltradosTemp = this.estabelecimentosFiltrados;
658     } else {
659         estabelecimentosFiltradosTemp = this.estabelecimentos;
660     }
661     ArrayList<CharSequence> tempLabelsTiposAtendimentos = new ArrayList<CharSequence>();
662     ArrayList<CharSequence> tempLabelsTiposProfissionais = new ArrayList<CharSequence>();
663     ArrayList<CharSequence> tempLabelsTiposConvenios = new ArrayList<CharSequence>();
664     ArrayList<CharSequence> tempLabelsTiposServicos = new ArrayList<CharSequence>();
665     ArrayList<CharSequence> tempLabelsTiposEstabelecimentos = new ArrayList<CharSequence>();
666     for (Estabelecimento e : estabelecimentosFiltradosTemp) {
667         if (!tempLabelsTiposEstabelecimentos.contains(e.getTipoEstabelecimento())) {
668             tempLabelsTiposEstabelecimentos.add(e.getTipoEstabelecimento());
669         }
670         for (Atendimento a : e.getAtendimento()) {
671             if (!tempLabelsTiposAtendimentos.contains(a.getTipoAtendimento())) {
672                 tempLabelsTiposAtendimentos.add(a.getTipoAtendimento());
673             }
674             if (!tempLabelsTiposConvenios.contains(a.getConvenio())) {
675                 tempLabelsTiposConvenios.add(a.getConvenio());
676             }
677         }
678         for (ServicoClassificacao sc : e.getServicos()) {
679             if (!tempLabelsTiposServicos.contains(sc.getDescricaoServico())) {
680                 tempLabelsTiposServicos.add(sc.getDescricaoServico());
681             }
682         }
683     }
684 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_\dimon\cadesaude\BuscaGPSActivity.java

```
681     }
682 }
683 for (Professional p : e.getProfissionais()) {
684     if (!tempLabelsTiposProfissionais.contains(p.getCboDescricao())) {
685         tempLabelsTiposProfissionais.add(p.getCboDescricao());
686     }
687 }
688 }
689 this.labelsTiposConvenios = tempLabelsTiposConvenios.toArray(new CharSequence[0]);
690 this.labelsTiposProfissionais = tempLabelsTiposProfissionais.toArray(new CharSequence[0]);
691 this.labelsTiposServicos = tempLabelsTiposServicos.toArray(new CharSequence[0]);
692 this.labelsTiposEstabelecimentos = tempLabelsTiposEstabelecimentos.toArray(new CharSequence[0]);
693 this.labelsTiposAtendimentos = tempLabelsTiposAtendimentos.toArray(new CharSequence[0]);
694 if (labelsTiposAtendimentos.length < 2) {
695     fabFiltroTipoAtend.setEnabled(false);
696 } else {
697     fabFiltroTipoAtend.setEnabled(true);
698 }
699 if (labelsTiposEstabelecimentos.length < 2) {
700     fabFiltroTipoEstab.setEnabled(false);
701 } else {
702     fabFiltroTipoEstab.setEnabled(true);
703 }
704 if (labelsTiposProfissionais.length < 2) {
705     fabFiltroTipoProf.setEnabled(false);
706 } else {
707     fabFiltroTipoProf.setEnabled(true);
708 }
709 if (labelsTiposServicos.length < 2) {
710     fabFiltroTipoServ.setEnabled(false);
711 } else {
712     fabFiltroTipoServ.setEnabled(true);
713 }
714 if (labelsTiposConvenios.length < 2) {
715     fabFiltroTipoConv.setEnabled(false);
716 }
```

```

716     } else {
717         fabFiltroTipoConv.setEnabled(true);
718     }
719     if (estabelecimentosFiltrados.size() == 1) {
720         fabFiltroTipoConv.setEnabled(false);
721         fabFiltroTipoServ.setEnabled(false);
722         fabFiltroTipoProf.setEnabled(false);
723         fabFiltroTipoEstab.setEnabled(false);
724         fabFiltroTipoAtend.setEnabled(false);
725     }
726     Arrays.sort(labelsTiposEstabelecimentos);
727     Arrays.sort(labelsTiposAtendimentos);
728     Arrays.sort(labelsTiposConvenios);
729     Arrays.sort(labelsTiposServicos);
730     Arrays.sort(labelsTiposProfissionais);
731 }
732
733 @Override
734 public void limparTodosFiltros() {
735     this.tiposServicosSelecionados.clear();
736     this.tiposProfissionaisSelecionados.clear();
737     this.tiposConveniosSelecionados.clear();
738     this.tiposEstabelecimentosSelecionados.clear();
739     this.tiposAtendimentosSelecionados.clear();
740     this.estabelecimentosFiltrados.clear();
741     meuAdapter = new EstabelecimentoAdapter(this, estabelecimentos, listaDeFavoritos, distanciaMaxima);
742     meuRecyclerView.setAdapter(meuAdapter);
743     filtrosAtivos.clear();
744     definirLabelsParaFiltros();
745     mostrarQtdeResultados();
746     botaoBusca.setVisible(true);
747     fabFiltroTipoEstab.setColorNormalResId(R.color.fab_normal);
748     fabFiltroTipoAtend.setColorNormalResId(R.color.fab_normal);
749     fabFiltroTipoConv.setColorNormalResId(R.color.fab_normal);
750     fabFiltroTipoServ.setColorNormalResId(R.color.fab_normal);

```

```

751 fabFiltroTipoProf.setColorNormalResId(R.color.fab_normal);
752 fabFiltroMenu.setMenuButtonColorNormalResId(R.color.fab_normal);
753 fabFiltroTipoEstab.setColorPressedResId(R.color.fab_pressed);
754 fabFiltroTipoAtend.setColorPressedResId(R.color.fab_pressed);
755 fabFiltroTipoConv.setColorPressedResId(R.color.fab_pressed);
756 fabFiltroTipoServ.setColorPressedResId(R.color.fab_pressed);
757 fabFiltroTipoProf.setColorPressedResId(R.color.fab_pressed);
758 fabFiltroMenu.setMenuButtonColorPressedResId(R.color.fab_pressed);
759 }
760
761 @Override
762 public void filtrarPorTipoDeConvenio(ArrayList<String> tiposConveniosSelecionados) {
763     this.tiposConveniosSelecionados = tiposConveniosSelecionados;
764     int compatibilidade = 0;
765     ArrayList<Estabelecimento> estabelecimentosTemp = null;
766     ArrayList<Estabelecimento> estabelecimentosTempFiltrados = new ArrayList<Estabelecimento>();
767     if (filtrosAtivos.size() > 0) {
768         estabelecimentosTemp = estabelecimentosFiltrados;
769     } else {
770         estabelecimentosTemp = estabelecimentos;
771     }
772     if (this.tiposConveniosSelecionados.size() > 0) {
773         for (Estabelecimento e : estabelecimentosTemp) {
774             for (Atendimento a : e.getAtendimento()) {
775                 if (this.tiposConveniosSelecionados.contains(a.getConvenio())) {
776                     compatibilidade++;
777                 }
778             }
779             if (compatibilidade > 0) {
780                 fabFiltroMenu.setMenuButtonColorPressedResId(R.color.accent);
781                 fabFiltroMenu.setMenuButtonColorNormalResId(R.color.primary);
782                 fabFiltroTipoConv.setColorNormalResId(R.color.primary);
783                 fabFiltroTipoConv.setColorPressedResId(R.color.accent);
784                 estabelecimentosTempFiltrados.add(e);
785                 compatibilidade = 0;

```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
786         }
787     }
788     meuAdapter = new EstabelecimentoAdapter(this, estabelecimentosTempFiltrados, listaDeFavoritos,
    distanciaMaxima);
789     meuRecyclerView.invalidate();
790     meuRecyclerView.setAdapter(meuAdapter);
791     if (!filtrosAtivos.contains(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_CONVENIO)) {
792         filtrosAtivos.add(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_CONVENIO);
793         botaoBusca.setVisible(false);
794     }
795     this.estabelecimentosFiltrados = estabelecimentosTempFiltrados;
796     definirLabelsParaFiltros();
797     mostrarQtdeResultados();
798 }
799
800
801 @Override
802 public void filtrarPorTipoDeProfissional(ArrayList<String> tiposProfissionaisSelecionados) {
803     this.tiposProfissionaisSelecionados = tiposProfissionaisSelecionados;
804     int compatibilidade = 0;
805     ArrayList<Estabelecimento> estabelecimentosTemp = null;
806     ArrayList<Estabelecimento> estabelecimentosTempFiltrados = new ArrayList<Estabelecimento>();
807     if (filtrosAtivos.size() > 0) {
808         estabelecimentosTemp = estabelecimentosFiltrados;
809     } else {
810         estabelecimentosTemp = estabelecimentos;
811     }
812     if (this.tiposProfissionaisSelecionados.size() > 0) {
813         for (Estabelecimento e : estabelecimentosTemp) {
814             for (Profissional p : e.getProfissionais()) {
815                 if (this.tiposProfissionaisSelecionados.contains(p.getCboDescricao())) {
816                     compatibilidade++;
817                 }
818             }
819         }
820         if (compatibilidade > 0) {
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
820     fabFiltroMenu.setMenuButtonColorPressedResId(R.color.accent);
821     fabFiltroMenu.setMenuButtonColorNormalResId(R.color.primary);
822     fabFiltroTipoProf.setColorNormalResId(R.color.primary);
823     fabFiltroTipoProf.setColorPressedResId(R.color.accent);
824     estabelecimentosTempFiltrados.add(e);
825     compatibilidade = 0;
826     }
827 }
828 meuAdapter = new EstabelecimentoAdapter(this, estabelecimentosTempFiltrados, listaDeFavoritos,
    distanciaMaxima);
829 meuRecyclerView.invalidate();
830 meuRecyclerView.setAdapter(meuAdapter);
831 if (!filtrosAtivos.contains(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_PROFISSIONAL)) {
832     filtrosAtivos.add(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_PROFISSIONAL);
833     botaoBusca.setVisible(false);
834 }
835 this.estabelecimentosFiltrados = estabelecimentosTempFiltrados;
836 definirLabelsParaFiltros();
837 mostrarQtdeResultados();
838 }
839 }
840
841 @Override
842 public void filtrarPorTipoDeServico(ArrayList<String> tiposServicosSelecionados) {
843     this.tiposServicosSelecionados = tiposServicosSelecionados;
844     int compatibilidade = 0;
845     ArrayList<Estabelecimento> estabelecimentosTemp = null;
846     ArrayList<Estabelecimento> estabelecimentosTempFiltrados = new ArrayList<Estabelecimento>();
847     if (filtrosAtivos.size() > 0) {
848         estabelecimentosTemp = estabelecimentosFiltrados;
849     } else {
850         estabelecimentosTemp = estabelecimentos;
851     }
852     if (this.tiposServicosSelecionados.size() > 0) {
853         for (Estabelecimento e : estabelecimentosTemp) {
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
854     for (ServicoClassificacao sc : e.getServicos()) {
855         if (this.tiposServicosSelecionados.contains(sc.getDescricaoServico())) {
856             compatibilidade++;
857         }
858     }
859     if (compatibilidade > 0) {
860
861         fabFiltroMenu.setMenuItemColorPressedResId(R.color.accent);
862         fabFiltroMenu.setMenuItemColorNormalResId(R.color.primary);
863         fabFiltroTipoServ.setMenuItemColorNormalResId(R.color.primary);
864         fabFiltroTipoServ.setMenuItemColorPressedResId(R.color.accent);
865         estabelecimentosTempFiltros.add(e);
866         compatibilidade = 0;
867     }
868 }
869 meuAdapter = new EstabelecimentoAdapter(this, estabelecimentosTempFiltros, listaDeFavoritos,
870     distanciaMaxima);
871 meuRecyclerView.invalidate();
872 meuRecyclerView.setAdapter(meuAdapter);
873 if (!filtrosAtivos.contains(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_SERVICO)) {
874     filtrosAtivos.add(InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_SERVICO);
875     botaoBusca.setVisible(false);
876 }
877 this.estabelecimentosFiltros = estabelecimentosTempFiltros;
878 definirLabelsParaFiltros();
879 mostraQtdeResultados();
880 }
881
882 @Override
883 protected void onResume() {
884     super.onResume();
885     if (meuRecyclerView != null) {
886         meuRecyclerView.setVisibility(View.GONE);
887     }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
888 fabFiltroMenu.setVisibility(View.GONE);
889 barraDeProgresso.setVisibility(View.VISIBLE);
890 }
891
892 @Override
893 public void googlePlayServiceIndisponivel() {
894     onBackPressed();
895 }
896
897 private class CarregadorDeDadosBGPSA extends AsyncTask<Void, Void, Void> {
898     private Context contexto; ArrayList<Profissional>
899     listaDeProfissionais;
900     ArrayList<ServicoClassificacao> listaDeServicos;
901     ArrayList<Atendimento> listaDeAtendimento;
902     ProfissionaisDAO profissionaisDAO;
903     ServicoClassificacaoDAO servicoClassificacaoDAO;
904     AtendimentoDAO atendimentoDAO;
905
906     public CarregadorDeDadosBGPSA(Context contexto) {
907         this.contexto = contexto;
908     }
909
910     @Override
911     protected void doInBackground(Void... params) {
912         Log.e("Estabelecimentos", "Carregando dados em background");
913         config = getSharedPreferences("favoritos", MODE_PRIVATE);
914         listaDeFavoritos = config.getStringSet("listaDeFavoritos", new HashSet<String>());
915         distanciaMaxima = String.valueOf(PreferenceManager.getDefaultSharedPreferences(contexto).getString(
916             R.string.pref_distancia_estabelecimentos key), 1));
917         if ((bkpEstabelecimentos == null || coordenadasAlteradas == true) && isCancelled() == false) {
918             Log.e("Cancelamento", "1");
919             estabDAO = EstabelecimentoDAO.getInstance(getApplicationContext());
920             if (isCancelled() == false) {
921                 Log.e("Cancelamento", "2");
922                 estabelecimentos = estabDAO.buscaEstabelecimentosPorGPS(localizacao, distanciaMaxima);
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_\dimon\cadesaude\BuscaGPSActivity.java

```
922     }
923     if (isCancelled() == false) {
924         Log.e("Cancelamento", "3");
925         Log.e("Estabelecimentos", "Estabelecimentos resultantes: " + String.valueOf(estabelecimentos.size()));
926     }
927     profissionaisDAO = ProfissionaisDAO.getInstance(getApplicationContext());
928     listaDeProfissionais = new ArrayList<Profissional>();
929     servicoClassificacaoDAO = ServicoClassificacaoDAO.getInstance(getApplicationContext());
930     listaDeServicos = new ArrayList<ServicoClassificacao>();
931     atendimentoDAO = AtendimentoDAO.getInstance(getApplicationContext());
932     listaDeAtendimento = new ArrayList<Atendimento>();
933 }
934 if (isCancelled() == false) {
935     Log.e("Cancelamento", "4");
936     for (Estabelecimento e : estabelecimentos) {
937         if (isCancelled()) {
938             Log.e("Cancelamento", "Cancelado no for de estabelecimentos");
939             break;
940         } else {
941             listaDeProfissionais = profissionaisDAO.listaProfissionais(String.valueOf(e.getCnes()));
942             listaDeServicos = servicoClassificacaoDAO.listaServicosEspecializados(String.valueOf(e.getCnes()));
943             listaDeAtendimento = atendimentoDAO.listaAtendimentos(String.valueOf(e.getCnes()));
944             e.setProfissionais(listaDeProfissionais);
945             e.setServicos(listaDeServicos);
946             e.setAtendimento(listaDeAtendimento);
947         }
948     }
949     bkpEstabelecimentos = estabelecimentos;
950     coordenadasAlteradas = false;
951 }
952 if (dadosDeBuscaNaTela.trim().length() > 0 && isCancelled() == false) {
953     Log.e("Cancelamento", "5");
954     dadosDeBuscaNaTela = StringUtils.removeAcentos(dadosDeBuscaNaTela);
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juceimar_dimon\cadesaude\BuscaGPSActivity.java

```
955     dadosDeBuscaNaTela = dadosDeBuscaNaTela.trim();
956     ArrayList<Estabelecimento> estabelecimentosTemp = new ArrayList<Estabelecimento>();
957     for (Estabelecimento e : bkpEstabelecimentos) {
958         if (e.getNomeFantasia().toLowerCase().contains(dadosDeBuscaNaTela.toLowerCase()) || e.
getRazaoSocial().toLowerCase().contains(dadosDeBuscaNaTela)) {
959             estabelecimentosTemp.add(e);
960         }
961     }
962     estabelecimentos = estabelecimentosTemp;
963     definirLabelsParaFiltros();
964     } else {
965         estabelecimentos = bkpEstabelecimentos;
966     }
967     return null;
968 }
969
970 @Override
971 protected void onCancelled(Void aVoid) {
972     super.onCancelled(aVoid);
973     if (estabelecimentos != null) {
974         Log.e("Cancelamento", String.valueOf(estabelecimentos.size()));
975         estabelecimentos.clear();
976     }
977     if (estabelecimentosFiltrados != null)
978         estabelecimentosFiltrados.clear();
979     if (bkpEstabelecimentos != null)
980         bkpEstabelecimentos.clear();
981     barraDeProgresso.setVisibility(View.GONE);
982     if (estabelecimentos != null)
983         Log.e("Cancelamento", String.valueOf(estabelecimentos.size()));
984     Log.e("Cancelamento", "onCancelledChamado");
985 }
986
987 @Override
988 protected void onPostExecute(Void aVoid) {
989     ...
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\BuscaGPSActivity.java

```
989     super.onPostExecute(aVoid);
990     Log.e("Cancelamento", "onPostExecuteChamado");
991     if (filtrosAtivos.size() > 0) {
992         botaoBusca.setVisibility(false);
993         Log.e("Estabelecimentos", String.valueOf(estabelecimentos.size()));
994         for (Integer i : filtrosAtivos) {
995             restaurarFiltros(i);
996         }
997     } else {
998         botaoBusca.setVisibility(true);
999         meuAdapter = new EstabelecimentoAdapter(contexto, estabelecimentos, listaDeFavoritos, distanciaMaxima
1000     );
1001     meuRecyclerView.setAdapter(meuAdapter);
1002     if (posicaoNaLista > -1) {
1003         meuRecyclerView.scrollToPosition(posicaoNaLista);
1004     }
1005     barraDeProgresso.setVisibility(View.GONE);
1006     fabFiltroMenu.setVisibility(View.VISIBLE);
1007     meuRecyclerView.setVisibility(View.VISIBLE);
1008     toolbar.setSubtitle("Raio de " + distanciaMaxima + " Km");
1009     fabFiltroMenu.showMenuButton(true);
1010     botaoGps.setVisibility(true);
1011     definirLabelsParaFiltros();
1012     mostraQtdeResultados();
1013 }
1014 }
1015 }
1016 }
1017 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\CaixaDeDialogoGoogleAPI.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.app.AlertDialog;
4 import android.app.Dialog;
5 import android.app.DialogFragment;
6 import android.content.DialogInterface;
7 import android.os.Bundle;
8
9 /**
10  * Created by Jucemar on 08/05/2016.
11  */
12 public class CaixaDeDialogoGoogleAPI extends DialogFragment implements DialogInterface.OnClickListener {
13
14     private ErroGooglePlayService erroGooglePlayService;
15
16     public void setErroGooglePlayService(ErroGooglePlayService erroGooglePlayService) {
17         this.erroGooglePlayService = erroGooglePlayService;
18     }
19
20     public CaixaDeDialogoGoogleAPI() {
21     }
22
23     @Override
24     public Dialog onCreateDialog(Bundle savedInstanceState) {
25         AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
26         builder.setTitle("Erro Google Play Service")
27             .setPositiveButton("Sair", this)
28             .setMessage("Erro ao tentar adquirir suas coordenadas geográficas. Não é possível continuar.");
29         return builder.create();
30     }
31
32     @Override
33     public void onClick(DialogInterface dialog, int which) {
34         erroGooglePlayService.googlePlayServiceIndisponivel();
35     }
36 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\CaixaDeDialogoGoogleAPI.java

```
36
37     public interface ErroGooglePlayService {
38         public void googlePlayServiceIndisponivel();
39     }
40 }
41
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Estabelecimento.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.os.Parcel;
4 import android.os.Parcelable;
5
6 import java.util.ArrayList;
7 import java.util.Comparator;
8
9 /**
10  * Created by Jucemar on 10/03/2016.
11  */
12 public class Estabelecimento implements Parcelable, Comparable<Float>, Comparable<Estabelecimento> {
13
14     private int cnes;
15     private String razaoSocial;
16     private String nomeFantasia;
17     private String logradouro;
18     private String numero;
19     private String complemento;
20     private String bairro;
21     private String cep;
22     private String municipio;
23     private String codigoMunicipio;
24     private String estado;
25     private String telefone;
26     private String tipoEstabelecimento;
27     private String latitude;
28     private Double longitude;
29     private Double distancia;
30     private ArrayList<Profissional> profissionais;
31     private ArrayList<ServicoClassificacao> servicos;
32     private ArrayList<Atendimento> atendimentos;
33     public static final String CNES = "cnes";
34     public static final String RAZAO_SOCIAL = "razaoSocial";
35     public static final String NOME_FANTASIA = "nomeFantasia";
36 }
```


File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\juemar_dimon\cadesaude\Estabelecimento.java

```
36 public static final String TELEFONE = "telefone";
37 public static final String LOGRADOURO = "logradouro";
38 public static final String NUMERO = "numero";
39 public static final String COMPLEMENTO = "complemento";
40 public static final String BAIRRO = "bairro";
41 public static final String MUNICIPIO = "municipio";
42 public static final String CODIGO_MUNICIPIO = "codigoMunicipio";
43 public static final String ESTADO = "estado";
44 public static final String CEP = "cep";
45 public static final String LATITUDE = "latitude";
46 public static final String LONGITUDE = "longitude";
47 public static final String TIPO_ESTABELECIMENTO = "tipoEstabelecimento";
48
49 public Estabelecimento() {
50 }
51
52 public Estabelecimento(int cnes, String razaoSocial, String nomeFantasia, String logradouro, String numero, String
complemento, String bairro, String cep, String municipio, String codigoMunicipio, String estado, String telefone,
String tipoEstabelecimento, Double latitude, Double longitude) {
53     this.cnes = cnes; this.razaoSocial
54     = razaoSocial; this.nomeFantasia =
55     nomeFantasia; this.logradouro =
56     logradouro; this.numero = numero;
57     this.complemento = complemento;
58     this.bairro = bairro;
59     this.cep = cep; this.municipio =
60     municipio; this.codigoMunicipio =
61     codigoMunicipio; this.estado = estado;
62     this.telefone = telefone;
63     this.tipoEstabelecimento = tipoEstabelecimento;
64     this.latitude = latitude;
65     this.longitude = longitude;
66
67 }
68 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Estabelecimento.java

```
69
70
71     protected Estabelecimento(Parcel in) {
72         cnas = in.readInt(); razaoSocial
73         = in.readString(); nomeFantasia
74         = in.readString(); logradouro =
75         in.readString(); numero =
76         in.readString(); complemento =
77         in.readString(); bairro =
78         in.readString();
79         cep = in.readString(); municipio =
80         in.readString(); codigoMunicipio =
81         in.readString(); estado =
82         in.readString();
83         telefone = in.readString();
84         tipoEstabelecimento = in.readString();
85         latitude = in.readDouble();
86         longitude = in.readDouble();
87         distancia = in.readFloat();
88     }
89
90     public static final Creator<Estabelecimento> CREATOR = new Creator<Estabelecimento>() {
91         @Override
92         public Estabelecimento createFromParcel(Parcel in) {
93             return new Estabelecimento(in);
94         }
95
96         @Override
97         public Estabelecimento[] newArray(int size) {
98             return new Estabelecimento[size];
99         }
100     };
101
102     @Override
103     public int describeContents() {
104         ...

```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Estabelecimento.java

```
104     return 0;
105 }
106
107 @Override
108 public void writeToParcel(Parcel dest, int flags) {
109     dest.writeInt(cnes);
110     dest.writeString(razaoSocial);
111     dest.writeString(nomeFantasia);
112     dest.writeString(logradouro);
113     dest.writeString(numero);
114     dest.writeString(complemento);
115     dest.writeString(bairro);
116     dest.writeString(cep);
117     dest.writeString(municipio);
118     dest.writeString(codigoMunicipio);
119     dest.writeString(estado);
120     dest.writeString(telefone);
121     dest.writeString(tipoEstabelecimento);
122     dest.writeDouble(latitude);
123     dest.writeDouble(longitude);
124     dest.writeFloat(distancia);
125 }
126
127 public int getCnes() {
128     return cnes;
129 }
130
131 public void setCnes(int cnes) {
132     this.cnes = cnes;
133 }
134
135 public String getRazaoSocial() {
136     return razaoSocial;
137 }
138 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Estabelecimento.java

```
139 public void setRazaoSocial(String razaoSocial) {
140     this.razaoSocial = razaoSocial;
141 }
142
143 public String getNomeFantasia() {
144     return nomeFantasia;
145 }
146
147 public void setNomeFantasia(String nomeFantasia) {
148     this.nomeFantasia = nomeFantasia;
149 }
150
151 public String getLogradouro() {
152     return logradouro;
153 }
154
155 public void setLogradouro(String logradouro) {
156     this.logradouro = logradouro;
157 }
158
159 public String getNumero() {
160     return numero;
161 }
162
163 public void setNumero(String numero) {
164     this.numero = numero;
165 }
166
167 public String getComplemento() {
168     return complemento;
169 }
170
171 public void setComplemento(String complemento) {
172     this.complemento = complemento;
173 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Estabelecimento.java

```
174
175     public String getBairro() {
176         return bairro;
177     }
178
179     public void setBairro(String bairro) {
180         this.bairro = bairro;
181     }
182
183     public String getCep() {
184         return cep;
185     }
186
187     public void setCep(String cep) {
188         this.cep = cep;
189     }
190
191     public String getMunicipio() {
192         return municipio;
193     }
194
195     public void setMunicipio(String municipio) {
196         this.municipio = municipio;
197     }
198
199     public String getCodigoMunicipio() {
200         return codigoMunicipio;
201     }
202
203     public void setCodigoMunicipio(String codigoMunicipio) {
204         this.codigoMunicipio = codigoMunicipio;
205     }
206
207     public String getEstado() {
208         return estado;
209     }
210 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Estabelecimento.java

```
209     }
210
211     public void setEstado(String estado) {
212         this.estado = estado;
213     }
214
215     public String getTelefone() {
216         return telefone;
217     }
218
219     public void setTelefone(String telefone) {
220         this.telefone = telefone;
221     }
222
223     public String getTipoEstabelecimento() {
224         return tipoEstabelecimento;
225     }
226
227     public void setTipoEstabelecimento(String tipoEstabelecimento) {
228         this.tipoEstabelecimento = tipoEstabelecimento;
229     }
230
231     public Double getLatitude() {
232         return latitude;
233     }
234
235     public void setLatitude(Double latitude) {
236         this.latitude = latitude;
237     }
238
239     public Double getLongitude() {
240         return longitude;
241     }
242
243     public void setLongitude(Double longitude) {
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Estabelecimento.java

```
244     this.longitude = longitude;
245 }
246
247 public float getDistancia() {
248     return distancia;
249 }
250
251 public void setDistancia(float distancia) {
252     this.distancia = distancia;
253 }
254
255 public ArrayList<Profissional> getProfissionais() {
256     return profissionais;
257 }
258
259 public void setProfissionais(ArrayList<Profissional> profissionais) {
260     this.profissionais = profissionais;
261 }
262
263 public ArrayList<ServicoClassificacao> getServicos() {
264     return servicos;
265 }
266
267 public void setServicos(ArrayList<ServicoClassificacao> servicos) {
268     this.servicos = servicos;
269 }
270
271 public ArrayList<Atendimento> getAtendimento() {
272     return atendimento;
273 }
274
275 public void setAtendimento(ArrayList<Atendimento> atendimento) {
276     this.atendimento = atendimento;
277 }
278 ---
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juceamar_dimon\cadesaude\Estabelecimento.java

```
279  @Override
280  public String toString() {
281      return String.valueOf(distancia);
282  }
283
284
285  @Override
286  public int compareTo(Estabelecimento another) {
287      return compare(this.distancia, another.distancia);
288  }
289
290  @Override
291  public int compare(Float lhs, Float rhs) {
292      return Float.compare(lhs, rhs);
293  }
294  }
295
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3  * Created by Jucemar on 10/03/2016.
4  */
5
6 import android.os.AsyncTask;
7 import android.os.Bundle;
8 import android.support.design.widget.TabLayout;
9 import android.support.v4.app.Fragment;
10 import android.support.v4.app.FragmentManager;
11 import android.support.v4.app.FragmentPagerAdapter;
12 import android.support.v4.view.ViewPager;
13 import android.support.v7.app.AppCompatActivity;
14 import android.support.v7.widget.Toolbar;
15 import android.view.MenuItem;
16 import android.view.View;
17 import android.widget.ProgressBar;
18 import android.widget.TextView;
19
20 import java.util.ArrayList;
21 import java.util.List;
22
23 public class EstabelecimentoActivity extends AppCompatActivity {
24
25     private Estabelecimento estabelecimento;
26     private Toolbar toolbar;
27     private TabLayout tabLayout;
28     private ViewPager viewPager;
29     private int cnes;
30     private ArrayList<Profissional> listaDeProfissionais;
31     private ArrayList<Atendimento> listaDeAtendimento;
32     private ArrayList<ServicoClassificacao> listaDeServicos;
33     private ProgressBar barraDeProgresso;
34     private TextView txvTituloAppBar;
```

```

36
37
38     @Override
39     protected void onCreate(Bundle savedInstanceState) {
40         super.onCreate(savedInstanceState);
41         setContentView(R.layout.activity_estabelecimento);
42         viewPager = (ViewPager) findViewById(R.id.viewpager);
43         tabLayout = (TabLayout) findViewById(R.id.tabs);
44         barraDeProgresso = (ProgressBar) findViewById(R.id.barra_progresso_estabelecimento_activity);
45         estabelecimento = getIntent().getParcelableExtra("estabelecimento");
46         txvTituloAppBar = (TextView) findViewById(R.id.titulo_estab_activity_estab);
47         cnes = estabelecimento.getCnes();
48         txvTituloAppBar.setText(estabelecimento.getNomeFantasia());
49         toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_estabelecimento);
50         setSupportActionBar(toolbar);
51         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
52         CarregadorDeDados carregadorDeDados = new CarregadorDeDados();
53         carregadorDeDados.execute();
54     }
55
56
57     private void configVisualizadorDePaginas(ViewPager viewPager) {
58         VisualizadorDePaginasAdapter adapter = new VisualizadorDePaginasAdapter(getSupportFragmentManager());
59         EstabelecimentoToFragment ef = new EstabelecimentoToFragment();
60         ef.setArguments(getIntent().getExtras());
61         ProfissionaisFragment pf = new ProfissionaisFragment(); Bundle bpf = new
62             Bundle(); bpf.putParcelableArrayList("listaDeProfissionais",
63             listaDeProfissionais); pf.setArguments(bpf);
64         ServicosFragment sf = new ServicosFragment();
65         Bundle bsf = new Bundle();
66         bsf.putParcelableArrayList("listaDeServicos", listaDeServicos);
67         sf.setArguments(bsf);
68         AtendimentoToFragment af = new AtendimentoToFragment();
69         Bundle baf = new Bundle();

```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoActivity.java

```
71 baf.putParcelableArrayList("ListaDeAtendimentos", listaDeAtendimento);
72 af.setArguments(baf);
73 adapter.addFragment(ef, "ESTABELECIMENTO");
74 adapter.addFragment(pf, "PROFISSIONAIS");
75 adapter.addFragment(sf, "SERVIÇOS");
76 adapter.addFragment(af, "ATENDIMENTO");
77 viewPager.setAdapter(adapter);
78 }
79
80
81 @Override
82 public boolean onOptionsItemSelected(MenuItem item) {
83     int id = item.getItemId();
84     switch (id) {
85         case android.R.id.home:
86             onBackPressed();
87             break;
88     }
89     return super.onOptionsItemSelected(item);
90 }
91
92
93 private class VisualizadorDePaginasAdapter extends FragmentStatePagerAdapter {
94     private final List<Fragment> mFragmentManager = new ArrayList<>();
95     private final List<String> mFragmentTitleList = new ArrayList<>();
96
97     public VisualizadorDePaginasAdapter(FragmentManager manager) {
98         super(manager);
99     }
100
101     @Override
102     public Fragment getItem(int position) {
103         return mFragmentManager.get(position);
104     }
105 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoActivity.java

```
106  @Override
107  public int getCount() {
108      return mFragmentManager.size();
109  }
110
111  public void addFragment(Fragment fragment, String title) {
112      mFragmentManager.add(fragment);
113      mFragmentManager.add(title);
114  }
115
116  @Override
117  public CharSequence getPageTitle(int position) {
118      return mFragmentManager.getTitle(position);
119  }
120  }
121
122  private class CarregadorDeDados extends AsyncTask<Void, Void, Void> {
123
124      @Override
125      protected void onPreExecute() {
126          super.onPreExecute();
127          barraDeProgresso.setVisibility(View.VISIBLE);
128          viewPager.setVisibility(View.GONE);
129      }
130
131      @Override
132      protected void doInBackground(Void... params) {
133          ProfissionaisDAO profissionaisDAO = ProfissionaisDAO.getInstance(getApplicationContext());
134          ListaDeProfissionais = profissionaisDAO.listaProfissionais(String.valueOf(cnes));
135          ServicoClassificacaoDAO servicoClassificacaoDAO = ServicoClassificacaoDAO.getInstance(getApplicationContext());
136          ListaDeServicos = servicoClassificacaoDAO.listarServicosEspecializados(String.valueOf(cnes));
137          AtendimentoDAO atendimentoDAO = AtendimentoDAO.getInstance(getApplicationContext());
138          ListaDeAtendimento = atendimentoDAO.listarAtendimentos(String.valueOf(cnes));
139          return null;
140      }
141  }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoActivity.java

```
140     }
141
142     @Override
143     protected void onPostExecute(Void aVoid) {
144         configVisualizadorDePaginas(viewPager);
145         tabLayout.setupWithViewPager(viewPager);
146         barraDeProgresso.setVisibility(View.GONE);
147         viewPager.setVisibility(View.VISIBLE);
148
149     }
150 }
151
152 }
153
154
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoAdapter.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.content.SharedPreferences;
5 import android.support.design.widget.Snackbar;
6 import android.support.v7.widget.RecyclerView;
7 import android.util.Log;
8 import android.view.LayoutInflater;
9 import android.view.View;
10 import android.view.ViewGroup;
11
12 import java.text.NumberFormat;
13 import java.util.ArrayList;
14 import java.util.Locale;
15 import java.util.Set;
16
17 /**
18  * Created by Jucemar on 08/03/2016.
19  */
20 public class EstabelecimentoAdapter extends RecyclerView.Adapter<EstabelecimentoViewHolder> {
21
22     private Context contexto;
23     private ArrayList<Estabelecimento> listaDeEstabelecimentos;
24     private Set<String> listaDeFavoritos;
25     private String distanciaMaxima;
26     private SharedPreferences preferencias;
27     private NumberFormat numberFormat;
28
29     public EstabelecimentoAdapter(Context contexto, ArrayList<Estabelecimento> listaDeEstabelecimentos, Set<String>
30         listaDeFavoritos) {
31         this.contexto = contexto;
32         this.listaDeEstabelecimentos = listaDeEstabelecimentos;
33         this.listaDeFavoritos = listaDeFavoritos;
34         this.numberFormat = NumberFormat.getInstance(new Locale("pt", "BR")); //para números
35         numberFormat.setMaximumFractionDigits(2);
36     }
37 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoAdapter.java

```
35     }
36
37     // Construtor de Adapter para montar a RecyclerView da BuscaGPSActivity
38     public EstabelecimentoAdapter(Context contexto, ArrayList<Estabelecimento> estabelecimentos, Set<String>
39         listaDeFavoritos, String distanciaMaxima) {
40         this.listaDeEstabelecimentos = estabelecimentos;
41         this.listaDeFavoritos = listaDeFavoritos;
42         this.distanciaMaxima = distanciaMaxima;
43         this.contexto = contexto;
44         this.numberFormat = NumberFormat.getInstance(new Locale("pt", "BR")); //para números
45         numberFormat.setMaximumFractionDigits(2);
46     }
47
48     @Override
49     public EstabelecimentoViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
50         View v = LayoutInflater.from(contexto).inflate(R.layout.item_lista_estabelecimentos, parent, false);
51         EstabelecimentoViewHolder vh = new EstabelecimentoViewHolder(contexto, v, listaDeEstabelecimentos, this);
52         return vh;
53     }
54
55     @Override
56     public void onBindViewHolder(EstabelecimentoViewHolder holder, int position) {
57         Estabelecimento e = listaDeEstabelecimentos.get(position);
58         holder.nomeFantasia.setText(e.getNomeFantasia());
59         holder.bairro.setText(e.getBairro());
60         holder.municipio.setText(e.getMunicipio());
61         if (e.getDistancia() > 0) {
62             holder.distancia.setText((numberFormat.format(e.getDistancia() / 1000.0f)) + " Km");
63         }
64         holder.tipoEstabelecimento.setText(e.getTipoEstabelecimento());
65         if (e.getTelefone().length() > 1) {
66             holder.iconeTelefone.setImageResource(R.drawable.ic_phone_grey600_24dp);
67             holder.iconeTelefone.setAlpha(1.00f);
68         } else {
69             holder.iconeTelefone.setAlpha(0.30f);
70         }
71     }
72 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoAdapter.java

```
69         holder.iconeTelefone.setEnabled(false);
70     }
71     Log.i("favoritos", listaDeFavoritos.toString());
72     if (listaDeFavoritos.contains(String.valueOf(e.getCnes()))) {
73         holder.iconeFavoritos.setImageResource(R.drawable.ic_star_grey600_24dp);
74     } else {
75         holder.iconeFavoritos.setImageResource(R.drawable.ic_star_outline_grey600_24dp);
76     }
77 }
78
79 @Override
80 public int getItemCount() {
81     return listaDeEstabelecimentos.size();
82 }
83
84
85
86 public void avisarRemocaoEmFavoritos(Estabelecimento estabelecimento) {
87     if (contexto instanceof MainActivity) {
88         listaDeEstabelecimentos.remove(estabelecimento);
89         notifyDataSetChanged();
90         ((MainActivity) contexto).checarFavoritos();
91     }
92     View v = null;
93     if (contexto instanceof MainActivity) {
94         v = ((MainActivity) contexto).findViewById(R.id.coordinator_layout_main_activity);
95     } else if (contexto instanceof BuscaGPSActivity) {
96         v = ((BuscaGPSActivity) contexto).findViewById(R.id.activity_busca_gps_layout);
97     } else {
98         v = ((BuscaGeralActivity) contexto).findViewById(R.id.activity_busca_geral_layout);
99     }
100     Snackbar.make(v, estabelecimento.getNomeFantasia() + " SAIU DOS FAVORITOS", Snackbar.LENGTH_SHORT).show();
101 }
102 public void avisarAdicaoDeFavoritos(Estabelecimento estabelecimento) {
103     ...
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoAdapter.java

```
104 View v = null;
105 if (contexto instanceof MainActivity) {
106     v = ((MainActivity) contexto).findViewById(R.id.coordinator_layout_main_activity);
107 } else if (contexto instanceof BuscaGPSActivity) {
108     v = ((BuscaGPSActivity) contexto).findViewById(R.id.activity_busca_gps_layout);
109 } else {
110     v = ((BuscaGeralActivity) contexto).findViewById(R.id.activity_busca_geral_layout);
111 }
112 Snackbar.make(v, estabelecimento.getNomeFantasia() + " AGORA É FAVORITO", Snackbar.LENGTH_SHORT).show();
113 }
114 }
115 }
116 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoAdapter.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.content.SharedPreferences;
5 import android.support.design.widget.Snackbar;
6 import android.support.v7.widget.RecyclerView;
7 import android.util.Log;
8 import android.view.LayoutInflater;
9 import android.view.View;
10 import android.view.ViewGroup;
11
12 import java.text.NumberFormat;
13 import java.util.ArrayList;
14 import java.util.Locale;
15 import java.util.Set;
16
17 /**
18  * Created by Jucemar on 08/03/2016.
19  */
20 public class EstabelecimentoAdapter extends RecyclerView.Adapter<EstabelecimentoViewHolder> {
21
22     private Context contexto;
23     private ArrayList<Estabelecimento> listaDeEstabelecimentos;
24     private Set<String> listaDeFavoritos;
25     private String distanciaMaxima;
26     private SharedPreferences preferencias;
27     private NumberFormat numberFormat;
28
29     public EstabelecimentoAdapter(Context contexto, ArrayList<Estabelecimento> listaDeEstabelecimentos, Set<String>
30         listaDeFavoritos) {
31         this.contexto = contexto;
32         this.listaDeEstabelecimentos = listaDeEstabelecimentos;
33         this.listaDeFavoritos = listaDeFavoritos;
34         this.numberFormat = NumberFormat.getInstance(new Locale("pt", "BR")); //para números
35         numberFormat.setMaximumFractionDigits(2);
36     }
37 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoAdapter.java

```
35     }
36
37     // Construtor de Adapter para montar a RecyclerView da BuscaGPSActivity
38     public EstabelecimentoAdapter(Context contexto, ArrayList<Estabelecimento> estabelecimentos, Set<String>
39         listaDeFavoritos, String distanciaMaxima) {
40         this.listaDeEstabelecimentos = estabelecimentos;
41         this.listaDeFavoritos = listaDeFavoritos;
42         this.distanciaMaxima = distanciaMaxima;
43         this.contexto = contexto;
44         this.numberFormat = NumberFormat.getInstance(new Locale("pt", "BR")); //para números
45         numberFormat.setMaximumFractionDigits(2);
46     }
47
48     @Override
49     public EstabelecimentoViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
50         View v = LayoutInflater.from(contexto).inflate(R.layout.item_lista_estabelecimentos, parent, false);
51         EstabelecimentoViewHolder vh = new EstabelecimentoViewHolder(contexto, v, listaDeEstabelecimentos, this);
52         return vh;
53     }
54
55     @Override
56     public void onBindViewHolder(EstabelecimentoViewHolder holder, int position) {
57         Estabelecimento e = listaDeEstabelecimentos.get(position);
58         holder.nomeFantasia.setText(e.getNomeFantasia());
59         holder.bairro.setText(e.getBairro());
60         holder.municipio.setText(e.getMunicipio());
61         if (e.getDistancia() > 0) {
62             holder.distancia.setText((numberFormat.format(e.getDistancia() / 1000.0f) + " Km"));
63         }
64         holder.tipoEstabelecimento.setText(e.getTipoEstabelecimento());
65         if (e.getTelefone().length() > 1) {
66             holder.iconeTelefone.setImageResource(R.drawable.ic_phone_grey600_24dp);
67             holder.iconeTelefone.setAlpha(1.00f);
68         } else {
69             holder.iconeTelefone.setAlpha(0.30f);
70         }
71     }
72 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoAdapter.java

```
69         holder.iconeTelefone.setEnabled(false);
70     }
71     Log.i("favoritos", listaDeFavoritos.toString());
72     if (listaDeFavoritos.contains(String.valueOf(e.getCnes()))) {
73         holder.iconeFavoritos.setImageResource(R.drawable.ic_star_grey600_24dp);
74     } else {
75         holder.iconeFavoritos.setImageResource(R.drawable.ic_star_outline_grey600_24dp);
76     }
77 }
78
79 @Override
80 public int getItemCount() {
81     return listaDeEstabelecimentos.size();
82 }
83
84
85
86 public void avisarRemocaoEmFavoritos(Estabelecimento estabelecimento) {
87     if (contexto instanceof MainActivity) {
88         listaDeEstabelecimentos.remove(estabelecimento);
89         notifyDataSetChanged();
90         ((MainActivity) contexto).checarFavoritos();
91     }
92     View v = null;
93     if (contexto instanceof MainActivity) {
94         v = ((MainActivity) contexto).findViewById(R.id.coordinator_layout_main_activity);
95     } else if (contexto instanceof BuscaGPSActivity) {
96         v = ((BuscaGPSActivity) contexto).findViewById(R.id.activity_busca_gps_layout);
97     } else {
98         v = ((BuscaGeralActivity) contexto).findViewById(R.id.activity_busca_geral_layout);
99     }
100     Snackbar.make(v, estabelecimento.getNomeFantasia() + " SAIU DOS FAVORITOS", Snackbar.LENGTH_SHORT).show();
101 }
102 public void avisarAdicaoDeFavoritos(Estabelecimento estabelecimento) {
103     ...
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoAdapter.java

```
104 View v = null;
105 if (contexto instanceof MainActivity) {
106     v = ((MainActivity) contexto).findViewById(R.id.coordinator_layout_main_activity);
107 } else if (contexto instanceof BuscaGPSActivity) {
108     v = ((BuscaGPSActivity) contexto).findViewById(R.id.activity_busca_gps_layout);
109 } else {
110     v = ((BuscaGeralActivity) contexto).findViewById(R.id.activity_busca_geral_layout);
111 }
112 Snackbar.make(v, estabelecimento.getNomeFantasia() + " AGORA É FAVORITO", Snackbar.LENGTH_SHORT).show();
113 }
114
115 }
116
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.location.Location;
7 import android.util.Log;
8
9 import java.util.ArrayList;
10 import java.util.Collections;
11 import java.util.Set;
12
13 /**
14  * Created by Jucemar on 10/03/2016.
15  */
16 public class EstabelecimentoDAO {
17
18     private static EstabelecimentoDAO instancia;
19     private SQLiteDatabase bd; String[] columnas = {
20         TabelaEstabelecimento.Colunas.CNES,
21         TabelaEstabelecimento.Colunas.RAZAO_SOCIAL,
22         TabelaEstabelecimento.Colunas.NOME_FANTASIA,
23         TabelaEstabelecimento.Colunas.TELEFONE,
24         TabelaEstabelecimento.Colunas.NOME_LOGRADOURO,
25         TabelaEstabelecimento.Colunas.NUMERO,
26         TabelaEstabelecimento.Colunas.COMPLEMENTO,
27         TabelaEstabelecimento.Colunas.BAIRRO,
28         TabelaEstabelecimento.Colunas.MUNICIPIO,
29         TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO,
30         TabelaEstabelecimento.Colunas.ESTADO,
31         TabelaEstabelecimento.Colunas.CEP,
32         TabelaEstabelecimento.Colunas.LATITUDE,
33         TabelaEstabelecimento.Colunas.LONGITUDE,
34         TabelaEstabelecimento.Colunas.TIPO_ESTABELECIMENTO};
35 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
36
37
38 public static EstabelecimentoDAO getInstancia(Context contexto) {
39     if (instancia == null) {
40         instancia = new EstabelecimentoDAO(contexto);
41     }
42     return instancia;
43 }
44
45 private EstabelecimentoDAO(Context contexto) {
46     BancoDeDados bancoDeDados = BancoDeDados.getInstancia(contexto);
47     bd = bancoDeDados.getReadableDatabase();
48 }
49
50
51 public ArrayList<Estabelecimento> listaTodosEstabelecimentos() {
52     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
53     Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, columnas, TabelaEstabelecimento.Colunas.LATITUDE + "<0
and " + TabelaEstabelecimento.Colunas.LONGITUDE + "<0 ", null, null, null, TabelaEstabelecimento.Colunas.NOME_FANTASIA)
;
54     if (c.moveToFirst()) {
55         do {
56             Estabelecimento e = extrairDoCursor(c);
57             resultado.add(e);
58         } while (c.moveToNext());
59     }
60     c.close();
61     return resultado;
62 }
63
64 public ArrayList<Estabelecimento> listaEstabelecimentos(Set<String> setMunicipios) {
65     StringBuffer filtro = new StringBuffer();
66     String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
67     for (int i = 0; i < municipios.length; i++) {
68         filtro.append("estabelecimento." + TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO + "=? ");
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
69         if (i < municipios.length - 1) {
70             filtro.append("OR ");
71         }
72     }
73
74     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
75     Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, columnas, filtro.toString(), municipios, null, null,
76         TabelaEstabelecimento.Columnas.NOME_FANTASIA);
77     if (c.moveToFirst()) {
78         do {
79             Estabelecimento e = extrairDoCursor(c);
80             resultado.add(e);
81         } while (c.moveToNext());
82     }
83     c.close();
84     return resultado;
85 }
86
87 public CharSequence[] listarTiposEstabelecimentos(Set<String> setMunicipios) {
88     CharSequence[] resultado = null;
89     if (setMunicipios.size() > 0) {
90         StringBuffer filtro = new StringBuffer();
91         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
92
93         for (int i = 0; i < municipios.length; i++) {
94             filtro.append(TabelaEstabelecimento.Columnas.CODIGO_MUNICIPIO + "=? ");
95             if (i < municipios.length - 1) {
96                 filtro.append("OR ");
97             }
98         }
99         String query = "SELECT DISTINCT tipo_estabelecimento FROM estabelecimento WHERE " + filtro + "ORDER BY
100         tipo_estabelecimento";
101         Cursor c = bd.rawQuery(query, municipios);
102         resultado = new CharSequence[c.getCount()];
```


File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
102     int i = 0;
103     if (c.moveToFirst()) {
104         do {
105             resultado[i] = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.TIPO_ESTABELECImento));
106             i++;
107             while (c.moveToNext());
108         }
109         c.close();
110     }
111     return resultado;
112 }
113
114
115 public ArrayList<Estabelecimento> listaEstabelecimentosFavoritos(Set<String> listaDeFavoritos) {
116     if (listaDeFavoritos.size() > 0) {
117         ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
118         if (listaDeFavoritos != null) {
119             StringBuffer filtro = new StringBuffer();
120             String[] favoritos = listaDeFavoritos.toArray(new String[listaDeFavoritos.size()]);
121             for (int i = 0; i < favoritos.length; i++) {
122                 filtro.append(TabelaEstabelecimento.Colunas.CNES + "==" + favoritos[i] + " ");
123                 if (i < favoritos.length - 1) {
124                     filtro.append("OR ");
125                 }
126             }
127             Log.e("Filtro", filtro.toString());
128             Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, colunas, filtro.toString(), null, null,
129                 TabelaEstabelecimento.Colunas.NOME_FANTASIA);
130             if (c.moveToFirst()) {
131                 do {
132                     Estabelecimento e = extrairDoCursor(c);
133                     resultado.add(e);
134                     while (c.moveToNext());
135                 }
136                 c.close();
137             }
138         }
139     }
140 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
136     }
137     return resultado;
138 } else {
139     return new ArrayList<Estabelecimento>();
140 }
141 }
142
143 public static final Estabelecimento extrairDoCursor(Cursor c) {
144     int cnes = c.getInt(c.getColumnIndex(TabelaEstabelecimento.Colunas.CNES));
145     String razaoSocial = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.RAZAO_SOCIAL));
146     String nomeFantasia = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.NOME_FANTASIA));
147     String telefone = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.TELEFONE));
148     String nomeLogradouro = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.NOME_LOGRADOURO));
149     String numero = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.NUMERO));
150     String complemento = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.COMPLEMENTO));
151     String bairro = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.BAIRRO)); municipio =
152     String c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.MUNICIPIO)); codigoMunicipio =
153     String c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO)); estado =
154     String c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.ESTADO));
155     String cep = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.CEP));
156     Double latitude = c.getDouble(c.getColumnIndex(TabelaEstabelecimento.Colunas.LATITUDE));
157     Double longitude = c.getDouble(c.getColumnIndex(TabelaEstabelecimento.Colunas.LONGITUDE));
158     String tipoEstabelecimento = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.TIPO_ESTABELECIMENTO));
159     ;
160     return new Estabelecimento(cnes, razaoSocial, nomeFantasia, nomeLogradouro, numero, complemento, bairro, cep,
161     municipio, codigoMunicipio, estado, telefone, tipoEstabelecimento, latitude, longitude);
162 }
163
164 public Estabelecimento getEstabelecimento(int cnes) {
165     Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, colunas, TabelaEstabelecimento.Colunas.CNES + "=" +
166     cnes, null, null, TabelaEstabelecimento.Colunas.NOME_FANTASIA);
167     c.moveToFirst();
168     Estabelecimento estabelecimento = extrairDoCursor(c);
169     c.close();
170     return estabelecimento;
171 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
168     }
169
170     public ArrayList<Estabelecimento> buscaStringEmEstabelecimentos(Set<String> setMunicipios, String sentenca) {
171         StringBuffer filtroDeMunicipios = new StringBuffer();
172         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
173         for (int i = 0; i < municipios.length; i++) {
174             filtroDeMunicipios.append("'" + municipios[i] + "'");
175             if (i < municipios.length - 1) {
176                 filtroDeMunicipios.append(", ");
177             }
178         }
179         String query = "SELECT * FROM estabelecimento WHERE (nome_fantasia LIKE '%" + sentenca + "%' OR razao_social LIKE '%" + sentenca + "%') AND codigo_municipio IN (" + filtroDeMunicipios.toString() + ")";
180         ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
181         Cursor c = bd.rawQuery(query, null);
182         if (c.moveToFirst()) {
183             do {
184                 Estabelecimento e = extrairDoCursor(c);
185                 resultado.add(e);
186             } while (c.moveToNext());
187         }
188         c.close();
189         return resultado;
190     }
191
192     public ArrayList<Estabelecimento> buscaEstabelecimentosPorGPS(Location localizacao, String distanciaMaxima) {
193         if (distanciaMaxima.length() == 0) {
194             distanciaMaxima = "0";
195         }
196         ArrayList<Estabelecimento> resultadoTemporario = new ArrayList<Estabelecimento>();
197         ArrayList<Estabelecimento> resultadoFinal = new ArrayList<Estabelecimento>();
198         resultadoTemporario = this.listaTodosEstabelecimentos();
199         for (Estabelecimento e : resultadoTemporario) {
200             Double lat = e.getLatitude();
201             Double lon = e.getLongitude();
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
202 float[] distancia = new float[1];
203 Location.distanceBetween(localizacao.getLatitudeO(), localizacao.getLongitudeO(), lat, lon, distancia);
204 if (distancia[0] <= Float.parseFloat(distanciaMaxima) * 1000 && lat != 0.0 && lon != 0.0) {
205     e.setDistancia(distancia[0]);
206     resultadoFinal.add(e);
207 }
208 }
209 Collections.sort(resultadoFinal);
210 for (Estabelecimento e : resultadoFinal) {
211     Log.e("Distâncias", e.getNomeFantasiaO() + " " + e.getDistanciaO());
212 }
213 return resultadoFinal;
214 }
215
216 public String buscaCodigoMunicipioAtual(Location localizacao) {
217     ArrayList<Estabelecimento> resultadoFinal = new ArrayList<Estabelecimento>();
218     resultadoFinal = this.listaTodosEstabelecimentosO();
219     for (Estabelecimento e : resultadoFinal) {
220         Double lat = e.getLatitudeO();
221         Double lon = e.getLongitudeO();
222         float[] distancia = new float[1];
223         Location.distanceBetween(localizacao.getLatitudeO(), localizacao.getLongitudeO(), lat, lon, distancia);
224         e.setDistancia(distancia[0]);
225     }
226     Collections.sort(resultadoFinal);
227     return resultadoFinal.get(0).getCodigoMunicipioO();
228 }
229 }
230
231
232
233
234
235
---
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.location.Location;
7 import android.util.Log;
8
9 import java.util.ArrayList;
10 import java.util.Collections;
11 import java.util.Set;
12
13 /**
14  * Created by Jucemar on 10/03/2016.
15  */
16 public class EstabelecimentoDAO {
17
18     private static EstabelecimentoDAO instancia;
19     private SQLiteDatabase bd; String[] columnas = {
20         TabelaEstabelecimento.Colunas.CNES,
21         TabelaEstabelecimento.Colunas.RAZAO_SOCIAL,
22         TabelaEstabelecimento.Colunas.NOME_FANTASIA,
23         TabelaEstabelecimento.Colunas.TELEFONE,
24         TabelaEstabelecimento.Colunas.NOME_LOGRADOURO,
25         TabelaEstabelecimento.Colunas.NUMERO,
26         TabelaEstabelecimento.Colunas.COMPLEMENTO,
27         TabelaEstabelecimento.Colunas.BAIRRO,
28         TabelaEstabelecimento.Colunas.MUNICIPIO,
29         TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO,
30         TabelaEstabelecimento.Colunas.ESTADO,
31         TabelaEstabelecimento.Colunas.CEP,
32         TabelaEstabelecimento.Colunas.LATITUDE,
33         TabelaEstabelecimento.Colunas.LONGITUDE,
34         TabelaEstabelecimento.Colunas.TIPO_ESTABELECIMENTO};
35 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
36
37
38 public static EstabelecimentoDAO getInstancia(Context contexto) {
39     if (instancia == null) {
40         instancia = new EstabelecimentoDAO(contexto);
41     }
42     return instancia;
43 }
44
45 private EstabelecimentoDAO(Context contexto) {
46     BancoDeDados bancoDeDados = BancoDeDados.getInstancia(contexto);
47     bd = bancoDeDados.getReadableDatabase();
48 }
49
50
51 public ArrayList<Estabelecimento> listaTodosEstabelecimentos() {
52     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
53     Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, colunas, TabelaEstabelecimento.Colunas.LATITUDE + "<0
and " + TabelaEstabelecimento.Colunas.LONGITUDE + "<0 ", null, null, null, TabelaEstabelecimento.Colunas.NOME_FANTASIA)
;
54     if (c.moveToFirst()) {
55         do {
56             Estabelecimento e = extrairDoCursor(c);
57             resultado.add(e);
58         } while (c.moveToNext());
59     }
60     c.close();
61     return resultado;
62 }
63
64 public ArrayList<Estabelecimento> listaEstabelecimentos(Set<String> setMunicipios) {
65     StringBuffer filtro = new StringBuffer();
66     String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
67     for (int i = 0; i < municipios.length; i++) {
68         filtro.append("estabelecimento." + TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO + "=? ");
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
69         if (i < municipios.length - 1) {
70             filtro.append("OR ");
71         }
72     }
73
74     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
75     Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, columnas, filtro.toString(), municipios, null, null,
76         TabelaEstabelecimento.Columnas.NOME_FANTASIA);
77     if (c.moveToFirst()) {
78         do {
79             Estabelecimento e = extrairDoCursor(c);
80             resultado.add(e);
81         } while (c.moveToNext());
82     }
83     c.close();
84     return resultado;
85 }
86
87 public CharSequence[] listarTiposEstabelecimentos(Set<String> setMunicipios) {
88     CharSequence[] resultado = null;
89     if (setMunicipios.size() > 0) {
90         StringBuffer filtro = new StringBuffer();
91         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
92
93         for (int i = 0; i < municipios.length; i++) {
94             filtro.append(TabelaEstabelecimento.Columnas.CODIGO_MUNICIPIO + "=? ");
95             if (i < municipios.length - 1) {
96                 filtro.append("OR ");
97             }
98         }
99         String query = "SELECT DISTINCT tipo_estabelecimento FROM estabelecimento WHERE " + filtro + "ORDER BY
100         tipo_estabelecimento";
101         Cursor c = bd.rawQuery(query, municipios);
102         resultado = new CharSequence[c.getCount()];
```


File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
102     int i = 0;
103     if (c.moveToFirst()) {
104         do {
105             resultado[i] = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.TIPO_ESTABELECIAMENTO));
106             i++;
107             while (c.moveToNext());
108         }
109         c.close();
110     }
111     return resultado;
112 }
113
114
115 public ArrayList<Estabelecimento> listaEstabelecimentosFavoritos(Set<String> listaDeFavoritos) {
116     if (listaDeFavoritos.size() > 0) {
117         ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
118         if (listaDeFavoritos != null) {
119             StringBuffer filtro = new StringBuffer();
120             String[] favoritos = listaDeFavoritos.toArray(new String[listaDeFavoritos.size()]);
121             for (int i = 0; i < favoritos.length; i++) {
122                 filtro.append(TabelaEstabelecimento.Colunas.CNES + "==" + favoritos[i] + " ");
123                 if (i < favoritos.length - 1) {
124                     filtro.append("OR ");
125                 }
126             }
127             Log.e("Filtro", filtro.toString());
128             Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, colunas, filtro.toString(), null, null,
129                 TabelaEstabelecimento.Colunas.NOME_FANTASIA);
130             if (c.moveToFirst()) {
131                 do {
132                     Estabelecimento e = extrairDoCursor(c);
133                     resultado.add(e);
134                 } while (c.moveToNext());
135             }
136             c.close();
137         }
138     }
139 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
136     }
137     return resultado;
138 } else {
139     return new ArrayList<Estabelecimento>();
140 }
141 }
142
143 public static final Estabelecimento extrairDoCursor(Cursor c) {
144     int cnes = c.getInt(c.getColumnIndex(TabelaEstabelecimento.Colunas.CNES));
145     String razaoSocial = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.RAZAO_SOCIAL));
146     String nomeFantasia = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.NOME_FANTASIA));
147     String telefone = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.TELEFONE));
148     String nomeLogradouro = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.NOME_LOGRADOURO));
149     String numero = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.NUMERO));
150     String complemento = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.COMPLEMENTO));
151     String bairro = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.BAIRRO)); municipio =
152     String c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.MUNICIPIO)); codigoMunicipio =
153     String c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO)); estado =
154     String c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.ESTADO));
155     String cep = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.CEP));
156     Double latitude = c.getDouble(c.getColumnIndex(TabelaEstabelecimento.Colunas.LATITUDE));
157     Double longitude = c.getDouble(c.getColumnIndex(TabelaEstabelecimento.Colunas.LONGITUDE));
158     String tipoEstabelecimento = c.getString(c.getColumnIndex(TabelaEstabelecimento.Colunas.TIPO_ESTABELECIMENTO));
159     ;
160     return new Estabelecimento(cnes, razaoSocial, nomeFantasia, nomeLogradouro, numero, complemento, bairro, cep,
161     municipio, codigoMunicipio, estado, telefone, tipoEstabelecimento, latitude, longitude);
162 }
163
164 public Estabelecimento getEstabelecimento(int cnes) {
165     Cursor c = bd.query(TabelaEstabelecimento.NOME_TABELA, colunas, TabelaEstabelecimento.Colunas.CNES + "=" +
166     cnes, null, null, TabelaEstabelecimento.Colunas.NOME_FANTASIA);
167     c.moveToFirst();
168     Estabelecimento estabelecimento = extrairDoCursor(c);
169     c.close();
170     return estabelecimento;
171 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\juceimar_dimon\cadesaude\EstabelecimentoDAO.java

```
168     }
169
170     public ArrayList<Estabelecimento> buscaStringEmEstabelecimentos(Set<String> setMunicipios, String sentenca) {
171         StringBuffer filtroDeMunicipios = new StringBuffer();
172         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
173         for (int i = 0; i < municipios.length; i++) {
174             filtroDeMunicipios.append("'" + municipios[i] + "'");
175             if (i < municipios.length - 1) {
176                 filtroDeMunicipios.append(", ");
177             }
178         }
179         String query = "SELECT * FROM estabelecimento WHERE (nome_fantasia LIKE '%" + sentenca + "%' OR razao_social LIKE '%" + sentenca + "%') AND codigo_municipio IN (" + filtroDeMunicipios.toString() + ")";
180         ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
181         Cursor c = bd.rawQuery(query, null);
182         if (c.moveToFirst()) {
183             do {
184                 Estabelecimento e = extrairDoCursor(c);
185                 resultado.add(e);
186             } while (c.moveToNext());
187         }
188         c.close();
189         return resultado;
190     }
191
192     public ArrayList<Estabelecimento> buscaEstabelecimentosPorGPS(Location localizacao, String distanciaMaxima) {
193         if (distanciaMaxima.length() == 0) {
194             distanciaMaxima = "0";
195         }
196         ArrayList<Estabelecimento> resultadoTemporario = new ArrayList<Estabelecimento>();
197         ArrayList<Estabelecimento> resultadoFinal = new ArrayList<Estabelecimento>();
198         resultadoTemporario = this.listaTodosEstabelecimentos();
199         for (Estabelecimento e : resultadoTemporario) {
200             Double lat = e.getLatitude();
201             Double lon = e.getLongitude();
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoDAO.java

```
202 float[] distancia = new float[1];
203 Location.distanceBetween(localizacao.getLatitudeO(), localizacao.getLongitudeO(), lat, lon, distancia);
204 if (distancia[0] <= Float.parseFloat(distanciaMaxima) * 1000 && lat != 0.0 && lon != 0.0) {
205     e.setDistancia(distancia[0]);
206     resultadoFinal.add(e);
207 }
208 }
209 Collections.sort(resultadoFinal);
210 for (Estabelecimento e : resultadoFinal) {
211     Log.e("Distâncias", e.getNomeFantasiaO() + " " + e.getDistanciaO());
212 }
213 return resultadoFinal;
214 }
215
216 public String buscaCodigoMunicipioAtual(Location localizacao) {
217     ArrayList<Estabelecimento> resultadoFinal = new ArrayList<Estabelecimento>();
218     resultadoFinal = this.listaTodosEstabelecimentosO();
219     for (Estabelecimento e : resultadoFinal) {
220         Double lat = e.getLatitudeO();
221         Double lon = e.getLongitudeO();
222         float[] distancia = new float[1];
223         Location.distanceBetween(localizacao.getLatitudeO(), localizacao.getLongitudeO(), lat, lon, distancia);
224         e.setDistancia(distancia[0]);
225     }
226     Collections.sort(resultadoFinal);
227     return resultadoFinal.get(0).getCodigoMunicipioO();
228 }
229 }
230 }
231
232
233
234
235
---
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoToFragment.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3  * Created by Jucemar on 18/03/2016.
4  */
5
6 import android.content.Intent;
7 import android.content.SharedPreferences;
8 import android.net.Uri;
9 import android.os.Bundle;
10 import android.support.v4.app.Fragment;
11 import android.util.Log;
12 import android.view.LayoutInflater;
13 import android.view.View;
14 import android.view.ViewGroup;
15 import android.widget.ImageButton;
16 import android.widget.TextView;
17
18 import com.github.clans.fab.FloatingActionButton;
19
20 import java.util.HashSet;
21 import java.util.Set;
22
23 public class EstabelecimentoToFragment extends Fragment implements View.OnClickListener {
24
25     private Estabelecimento estabelecimento;
26     private TextView txvRazaoSocial;
27     private TextView txvNomeFantasia;
28     private TextView txvLogradouro;
29     private TextView txvNumero;
30     private TextView txvComplemento;
31     private TextView txvBairro;
32     private TextView txvCep;
33     private TextView txvMunicipio;
34     private TextView txvEstado;
35     private TextView txvTelefone;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoFragment.java

```
36 private TextView txtTipoEstabelecimento;
37 private ImageButton iconeEndereco; ImageButton
38 private iconeTelefone; SharedPreferences
39 private config; SharedPreferences.Editor
40 private editorConfig; Set<String>
41 private listaDefavoritos; FloatingActionButton
42 private fabFavorito;
43
44 public EstabelecimentoFragment() {
45     // Required empty public constructor
46 }
47
48 @Override
49 public void onCreate(Bundle savedInstanceState) {
50     super.onCreate(savedInstanceState);
51     listaDefavoritos = new HashSet<String>();
52     Log.e("ExecutandoEmBackground", "Executando carregamento em background");
53     config = getActivity().getSharedPreferences("favoritos", getActivity().MODE_PRIVATE);
54     editorConfig = config.edit();
55     listaDefavoritos = config.getStringSet("listaDefavoritos", new HashSet<String>());
56 }
57
58 private void configurarFABs() {
59     if (listaDefavoritos.contains(String.valueOf(estabelecimento.getCnes()))) {
60         fabFavorito.setImageResource(R.drawable.ic_star_white_24dp);
61     } else {
62         fabFavorito.setImageResource(R.drawable.ic_star_outline_white_24dp);
63     }
64 }
65
66 @Override
67 public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
68     View v = inflater.inflate(R.layout.fragment_estabelecimento, container, false); estabelecimento
69     = getArguments().getParcelable("estabelecimento");
70     fabFavorito = (FloatingActionButton) v.findViewById(R.id.fab_add_favorito);
71 }
```

```

71 fabFavorito.setOnClickListener(this);
72 iconeEndereco = (ImageButton) v.findViewById(R.id.icone_endereco);
73 iconeEndereco.setOnClickListener(this);
74 iconeTelefone = (ImageButton) v.findViewById(R.id.icone_telefone);
75 iconeTelefone.setOnClickListener(this);
76 if (estabelecimento.getTelefone().length() < 4) {
77     iconeTelefone.setEnabled(false);
78     iconeTelefone.setAlpha(0.30f);
79 } else {
80     iconeTelefone.setEnabled(true);
81     iconeTelefone.setAlpha(1.00f);
82 }
83 txvRazaoSocial = (TextView) v.findViewById(R.id.label_razao_social_abrir_estabelecimento);
84 txvNomeFantasia = (TextView) v.findViewById(R.id.label_nome_fantasia_abrir_estabelecimento);
85 txvLogradouro = (TextView) v.findViewById(R.id.label_logradouro_abrir_estabelecimento);
86 txvNumero = (TextView) v.findViewById(R.id.label_numero_abrir_estabelecimento);
87 txvComplemento = (TextView) v.findViewById(R.id.label_complemento_abrir_estabelecimento);
88 txvBairro = (TextView) v.findViewById(R.id.label_bairro_abrir_estabelecimento);
89 txvCep = (TextView) v.findViewById(R.id.label_cep_abrir_estabelecimento);
90 txvMunicipio = (TextView) v.findViewById(R.id.label_municipio_abrir_estabelecimento);
91 txvEstado = (TextView) v.findViewById(R.id.label_estado_abrir_estabelecimento);
92 txvTelefone = (TextView) v.findViewById(R.id.label_telefone_abrir_estabelecimento);
93 txvTipoEstabelecimento = (TextView) v.findViewById(R.id.label_tipo_estabelecimento_abrir_estabelecimento);
94 txvBairro.setText(estabelecimento.getBairro());
95 txvRazaoSocial.setText(estabelecimento.getRazaoSocial().toString());
96 txvNomeFantasia.setText(estabelecimento.getNomeFantasia());
97 txvLogradouro.setText(estabelecimento.getLogradouro());
98 txvNumero.setText(estabelecimento.getNumero());
99 txvComplemento.setText(estabelecimento.getComplemento());
100 txvCep.setText(estabelecimento.getCep());
101 txvMunicipio.setText(estabelecimento.getMunicipio());
102 txvEstado.setText(estabelecimento.getEstado());
103 txvTelefone.setText(estabelecimento.getTelefone());
104 txvTipoEstabelecimento.setText(estabelecimento.getTipoEstabelecimento());

```


File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoToFragment.java

```
106     return v;
107 }
108
109 @Override
110 public void onClick(View v) {
111     int id = v.getId();
112     switch (id) {
113         case R.id.fab_add_favorito:
114             atualizaFavoritos();
115             break;
116         case R.id.icone_endereco:
117             abrirMapa();
118             break;
119         case R.id.icone_telefone:
120             fazerLigacao();
121             break;
122     }
123     Log.e("ListaDeFavoritosEst", listaDefavoritos.toString());
124 }
125
126 private void fazerLigacao() {
127     Intent intent = new Intent(Intent.ACTION_DIAL);
128     String telefone = StringUtils.formatarTelefones(estabelecimento.getTelefone());
129     intent.setData(Uri.parse("tel:" +
130         telefone));
131     getActivity().startActivity(intent);
132 }
133
134 private void abrirMapa() {
135     Double lat = estabelecimento.getLatitude();
136     Double lon = estabelecimento.getLongitude();
137     String nomeFantasia = estabelecimento.getNomeFantasia();
138     String rua = estabelecimento.getLogradouro();
139     String numero = estabelecimento.getNumero();
140     String cidade = estabelecimento.getMunicipio();
141     String estado = estabelecimento.getEstado();
142 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoFragment.java

```
141 Uri uri = null;
142 if (lat != 0.0 && lon != 0.0) {
143     String uriInicio = "geo:" + lat + "," + lon;
144     String query = lat + "," + lon + "(" + nomeFantasia + ")";
145     String encodedQuery = Uri.encode(query);
146     String uriString = uriInicio + "?q=" + encodedQuery + "&z=16";
147     uri = Uri.parse(uriString);
148 } else {
149     String uriInicio = "geo:" + 0 + "," + 0;
150     String uriString = uriInicio + "?q=" + rua + "%20" + numero + "%20" + cidade + "%2d" + estado;
151     uri = Uri.parse(uriString);
152 }
153 Log.e("endereço", uri.toString());
154 Intent intent = new Intent(android.content.Intent.ACTION_VIEW, uri);
155 startActivity(intent);
156 }
157
158 public void atualizaFavoritos() {
159     if (listaDefavoritos.contains(String.valueOf(estabelecimento.getCnes()))) {
160         listaDefavoritos.remove(String.valueOf(estabelecimento.getCnes()));
161         fabFavorito.setImageResource(R.drawable.ic_star_outline_white_24dp);
162     } else {
163         listaDefavoritos.add(String.valueOf(estabelecimento.getCnes()));
164         fabFavorito.setImageResource(R.drawable.ic_star_white_24dp);
165     }
166     editorConfig.putString("listaDefavoritos", listaDefavoritos);
167     editorConfig.apply();
168     Log.e("Favoritos", "Resultado: " + config.getStringSet("listaDefavoritos", null).toString());
169 }
170 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoViewHolder.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.net.Uri;
7 import android.support.v7.widget.RecyclerView;
8 import android.util.Log;
9 import android.view.View;
10 import android.widget.ImageButton;
11 import android.widget.TextView;
12
13 import java.util.ArrayList;
14 import java.util.HashSet;
15 import java.util.Set;
16
17 /**
18  * Created by Jucemar on 08/03/2016.
19  */
20 public class EstabelecimentoViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {
21
22     private Context context;
23     public TextView nomeFantasia;
24     public TextView tipoEstabelecimento;
25     public TextView bairro;
26     public TextView municipio;
27     ArrayList<Estabelecimento> listaEstabelecimentos;
28     public ImageButton iconeFavoritos;
29     public ImageButton iconeTelefone;
30     public ImageButton iconeGPS;
31     public TextView distancia;
32     private SharedPreferences config;
33     private SharedPreferences.Editor editorConfig;
34     private Set<String> listaDefavoritos;
35     private EstabelecimentoAdapter estabeleceAdapter;
36 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoViewHolder.java

```
36
37     public EstabelecimentoViewHolder(Context context, View itemView, ArrayList<Estabelecimento> listaEstabelecimentos,
EstabelecimentoAdapter estabelecimentosAdapter) {
38         super(itemView);
39         this.estabelecimentosAdapter = estabelecimentosAdapter;
40         this.listaEstabelecimentos = listaEstabelecimentos;
41         itemView.setOnClickListener(this);
42         this.context = context;
43         this.nomeFantasia = (TextView) itemView.findViewById(R.id.nome_fantasia_lista_busca);
44         this.tipoEstabelecimento = (TextView) itemView.findViewById(R.id.tipo_estab_lista_busca);
45         this.bairro = (TextView) itemView.findViewById(R.id.bairro_lista_busca);
46         this.municipio = (TextView) itemView.findViewById(R.id.municipio_lista_busca);
47         this.distancia = (TextView) itemView.findViewById(R.id.distancia_lista_busca);
48         this.iconeFavoritos = (ImageButton) itemView.findViewById(R.id.flag_favorito);
49         this.iconeTelefone = (ImageButton) itemView.findViewById(R.id.flag_fone);
50         this.iconeGPS = (ImageButton) itemView.findViewById(R.id.flag_gps);
51         this.iconeFavoritos.setOnClickListener(this);
52         this.iconeTelefone.setOnClickListener(this);
53         this.iconeGPS.setOnClickListener(this);
54         config = context.getSharedPreferences("favoritos", context.MODE_PRIVATE);
55         editorConfig = config.edit();
56         listaDefavoritos = config.getStringSet("listaDefavoritos", new HashSet<String>());
57     },
58
59     @Override
60     public void onClick(View v) {
61         Log.e("Click", String.valueOf(listaEstabelecimentos.get(getAdapterPosition()).getRazaoSocial()));
62         Estabelecimento e = listaEstabelecimentos.get(getAdapterPosition());
63         Intent i = new Intent(context, EstabelecimentoActivity.class);
64         i.putExtra("estabelecimento", e);
65         i.putExtra("listaPosicao", getAdapterPosition());
66         switch (v.getId()) {
67             case R.id.flag_favorito:
68                 toggleFavorito(e);
69         }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoViewHolder.java

```
70         break;
71     case R.id.flag_fone:
72         ligarParaEstabelecimento(e);
73         break;
74     case R.id.flag_gps:
75         abrirMapa(e);
76         break;
77     default:
78         context.startActivity(i);
79     }
80 }
81
82 private void toggleFavorito(Estabelecimento estabelecimento) {
83     if (listaDefavoritos.contains(String.valueOf(estabelecimento.getCnes()))) {
84         listaDefavoritos.remove(String.valueOf(estabelecimento.getCnes()));
85         this.iconeFavoritos.setImageResource(R.drawable.ic_star_outline_grey600_24dp);
86         estabelecimentosAdapter.avisarRemocaoEmFavoritos(estabelecimento);
87     } else {
88         listaDefavoritos.add(String.valueOf(estabelecimento.getCnes()));
89         this.iconeFavoritos.setImageResource(R.drawable.ic_star_grey600_24dp);
90         estabelecimentosAdapter.avisarAdicaoDefavoritos(estabelecimento);
91     }
92     Log.e("favoritos", String.valueOf(listaDefavoritos));
93     editorConfig.putStringSet("listaDefavoritos", listaDefavoritos);
94     editorConfig.apply();
95 }
96
97 private void ligarParaEstabelecimento(Estabelecimento estabelecimento) {
98     Intent intent = new Intent(Intent.ACTION_DIAL);
99     String telefone = StringUtils.formatarTelefones(estabelecimento.getTelefone());
100     intent.setData(Uri.parse("tel:" + telefone));
101     context.startActivity(intent);
102 }
103
104 private void abrirMapa(Estabelecimento estabelecimento) {
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\EstabelecimentoViewHolder.java

```
105 Double lat = estabelecimento.getLatitude();
106 Double lon = estabelecimento.getLongitude();
107 String nomeFantasia = estabelecimento.getNomeFantasia();
108 String rua = estabelecimento.getLogradouro();
109 String numero = estabelecimento.getNumero();
110 String cidade = estabelecimento.getMunicipio();
111 String estado = estabelecimento.getEstado();
112 Uri uri = null;
113 if (lat != 0.0 && lon != 0.0) {
114     String uriInicio = "geo:" + lat + "," + lon;
115     String query = lat + "," + lon + "(" + nomeFantasia + ")";
116     String encodedQuery = Uri.encode(query);
117     String uriString = uriInicio + "?q=" + encodedQuery + "&z=16";
118     uri = Uri.parse(uriString);
119 } else {
120     String uriInicio = "geo:" + 0 + "," + 0;
121     String uriString = uriInicio + "?q=" + rua + "%2C" + numero + "%20" + cidade + "%2d" + estado;
122     uri = Uri.parse(uriString);
123 }
124 Log.e("endereço", uri.toString());
125 Intent intent = new Intent(android.content.Intent.ACTION_VIEW, uri);
126 context.startActivity(intent);
127 }
128 }
129 }
130 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juccemar_dimon\cadesaude\FiltroDeEstabelecimentos.java

```
1 package br.com.juccemar_dimon.cadesaude;
2
3 import java.util.ArrayList;
4
5 /**
6  * Created by Juccemar on 04/04/2016.
7  */
8 public class FiltroDeEstabelecimentos {
9
10     public void filtrarPorTipoDeEstabelecimento(ArrayList<String> tiposEstabelecimentosSelecionados) {
11     }
12
13     public void filtrarPorTipoDeAtendimento(ArrayList<String> tiposAtendimentosSelecionados) {
14     }
15
16     public void filtrarPorTipoDeConvenio(ArrayList<String> tiposConveniosSelecionados) {
17     }
18
19     public void filtrarPorTipoDeProfissional(ArrayList<String> tiposProfissionaisSelecionados) {
20     }
21
22     public void filtrarPorTipoDeServico(ArrayList<String> tiposServicosSelecionados) {
23     }
24 }
25
26
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\FiltroGeralDialog.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.app.Activity;
4 import android.app.Dialog;
5 import android.app.DialogFragment;
6 import android.content.DialogInterface;
7 import android.os.Bundle;
8 import android.support.v7.app.AlertDialog;
9 import android.util.Log;
10 import android.widget.Button;
11
12 import java.util.ArrayList;
13
14 /**
15  * Created by Jucemar on 26/03/2016.
16  */
17 public class FiltroGeralDialog extends DialogFragment implements DialogInterface.OnClickListener,
    DialogInterface.OnClickListener {
18
19     private ArrayList<String> tiposSelecionados;
20     private ArrayList<String> bkpTiposSelecionados;
21     private InterfaceFiltroDeEstabelecimentos filtro;
22     private boolean[] itensChecados;
23     private CharSequence[] labelTipos;
24     private Button buttonConfirmar;
25     private Button buttonCancelar;
26     private int tipoDeDialog;
27     private AlertDialog dialog;
28     private int icone;
29     private ArrayList<String> tiposEstabelecimentosSelecionados;
30     private ArrayList<String> tiposAtendimentosSelecionados;
31     private ArrayList<String> tiposConveniosSelecionados;
32     private ArrayList<String> tiposProfissionaisSelecionados;
33     private ArrayList<String> tiposServicosSelecionados;
34 }
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\FiltroGeralDialog.java

```
35 public FiltroGeralDialog() {
36
37 }
38
39 public void setArgumentos(ArrayList<String> tiposSelecionados, int tipoDeDialog, CharSequence[] listaLabelTipos) {
40     this.labelTipos = listaLabelTipos;
41     switch (tipoDeDialog) {
42         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ATENDIMENTO:
43             tiposAtendimentosSelecionados = tiposSelecionados;
44             break;
45         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_CONVENIO:
46             tiposConveniosSelecionados = tiposSelecionados;
47             break;
48         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO:
49             tiposEstabelecimentosSelecionados = tiposSelecionados;
50             break;
51         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_PROFISSIONAL:
52             tiposProfissionaisSelecionados = tiposSelecionados;
53             break;
54         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_SERVICO:
55             tiposServicosSelecionados = tiposSelecionados;
56     }
57     this.tipoDeDialog = tipoDeDialog;
58     if (tiposSelecionados != null) {
59         this.tiposSelecionados = tiposSelecionados;
60     } else {
61         this.tiposSelecionados = new ArrayList<String>();
62     }
63     ,
64     @Override
65     public Dialog onCreateDialog(Bundle savedInstanceState) {
66         itensChecados = new boolean[labelTipos.length];
67         for (int i = 0; i < itensChecados.length; i++) {
68             itensChecados[i] = this.tiposSelecionados.contains(labelTipos[i]);
69         }
70     }
71 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\FiltroGeralDialog.java

```
70     }
71     AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
72     builder.setTitle(getString(R.string.titulo_dialog_filtro))
73         .setPositiveButton("Confirmar", this)
74         .setNegativeButton("Cancelar", this)
75         .setMultiChoiceItems(labelTipos, itensChecados, this);
76     this.dialog = builder.create();
77     return dialog;
78 }
79
80 @Override
81 public void onAttach(Activity activity) {
82     super.onAttach(activity);
83     if (!(activity instanceof InterfaceFiltroDeEstabelecimentos)) {
84         throw new RuntimeException("A activity precisa implementar a interface DialogFiltroTipoEstabelecimento.
85         InterfaceFiltroDeEstabelecimentos");
86     }
87     filtro = (InterfaceFiltroDeEstabelecimentos) activity;
88 }
89
90 private void addOnClick(int which, boolean isChecked) {
91     switch (tipoDeDialog) {
92         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ATENDIMENTO:
93             if (isChecked) {
94                 tiposAtendimentosSelecionados.add((String) labelTipos[which]);
95             } else {
96                 tiposAtendimentosSelecionados.remove((String) labelTipos[which]);
97             }
98             if (tiposAtendimentosSelecionados.size() > 0) {
99                 buttonConfirmar.setEnabled(true);
100             } else {
101                 buttonConfirmar.setEnabled(false);
102             }
103             break;
104         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_CONVENIO:
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\FiltroGeralDialog.java

```
104     if (isChecked) {
105         tiposConveniosSelecionados.add((String) labelTipos[which]);
106     } else {
107         tiposConveniosSelecionados.remove((String) labelTipos[which]);
108     }
109     if (tiposConveniosSelecionados.size() > 0) {
110         buttonConfirmar.setEnabled(true);
111     } else {
112         buttonConfirmar.setEnabled(false);
113     }
114     break;
115     case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO:
116     if (isChecked) {
117         tiposEstabelecimentosSelecionados.add((String) labelTipos[which]);
118     } else {
119         tiposEstabelecimentosSelecionados.remove((String) labelTipos[which]);
120     }
121     if (tiposEstabelecimentosSelecionados.size() > 0) {
122         buttonConfirmar.setEnabled(true);
123     } else {
124         buttonConfirmar.setEnabled(false);
125     }
126     break;
127     case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_PROFISSIONAL:
128     if (isChecked) {
129         tiposProfissionaisSelecionados.add((String) labelTipos[which]);
130     } else {
131         tiposProfissionaisSelecionados.remove((String) labelTipos[which]);
132     }
133     if (tiposProfissionaisSelecionados.size() > 0) {
134         buttonConfirmar.setEnabled(true);
135     } else {
136         buttonConfirmar.setEnabled(false);
137     }
138     .
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\FiltroGeralDialog.java

```
139     case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_SERVICO:
140         if (isChecked) {
141             tiposServicosSelecionados.add((String) labelTipos[which]);
142         } else {
143             tiposServicosSelecionados.remove((String) labelTipos[which]);
144         }
145         if (tiposServicosSelecionados.size() > 0) {
146             buttonConfirmar.setEnabled(true);
147         } else {
148             buttonConfirmar.setEnabled(false);
149         }
150     }
151 }
152
153 @Override
154 public void onClick(DialogInterface dialog, int which, boolean isChecked) {
155     addOnClick(which, isChecked);
156 }
157
158 @Override
159 public void onClick(DialogInterface dialog, int which) {
160     switch (which) {
161         case DialogInterface.BUTTON_POSITIVE:
162             escolherFiltro();
163             Log.e("Dialog", "Botão positivo clicado");
164             break;
165         case DialogInterface.BUTTON_NEGATIVE:
166             Log.e("Dialog", "Botão negativo clicado");
167             break;
168     }
169 }
170
171 private void escolherFiltro() {
172     switch (tipoDeDialog) {
173         case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ATENDIMENTO:
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\FiltroGeralDialog.java

```
174     filtro.filtrarPorTipoDeAtendimento(tiposAtendimentosSelecionados);
175     break;
176     case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_CONVENIO:
177         filtro.filtrarPorTipoDeConvenio(tiposConveniosSelecionados);
178         break;
179     case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_ESTABELECIMENTO:
180         filtro.filtrarPorTipoDeEstabelecimento(tiposEstabelecimentosSelecionados);
181         break;
182     case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_PROFISSIONAL:
183         filtro.filtrarPorTipoDeProfissional(tiposProfissionaisSelecionados);
184         break;
185     case InterfaceFiltroDeEstabelecimentos.FILTRO_TIPO_SERVICO:
186         filtro.filtrarPorTipoDeServico(tiposServicosSelecionados);
187         break;
188     }
189 }
190
191 @Override
192 public void onStart() { super.onStart();
193     dialog.setCanceledOnTouchOutside(false);
194     dialog.setCancelable(false);
195     this.buttonConfirmar = ((AlertDialog) dialog).getButton(DialogInterface.BUTTON_POSITIVE);
196     this.buttonCancelar = ((AlertDialog) dialog).getButton(DialogInterface.BUTTON_NEGATIVE);
197     if (tiposSelecionados.size() == 0) {
198         buttonConfirmar.setEnabled(false);
199     }
200 }
201
202 }
203
204 ~~~
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\InfoPreference.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3  * Created by Jucemar on 04/05/2016.
4  */
5
6 import android.content.Context;
7 import android.preference.Preference;
8 import android.util.AttributeSet;
9 import android.view.LayoutInflater;
10 import android.view.View;
11 import android.view.ViewGroup;
12 import android.widget.ImageView;
13 import android.widget.TextView;
14
15 public class InfoPreference extends Preference {
16     private TextView mensagem;
17     private ImageView icone;
18     private View viewDaPreferencia;
19
20     public TextView getMensagem() {
21         return mensagem;
22     }
23
24     public InfoPreference(Context context) {
25         super(context);
26     }
27
28     public InfoPreference(Context context, AttributeSet attrs) {
29         super(context, attrs);
30     }
31
32     public InfoPreference(Context context, AttributeSet attrs, int defStyleAttr) {
33         super(context, attrs, defStyleAttr);
34     }
35 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\InfoPreference.java

```
36  @Override
37  protected View onCreateView(ViewGroup parent) {
38      super.onCreateView(parent);
39      LayoutInflater inflater = (LayoutInflater) getContext().getSystemService(Context.LAYOUT_INFLATER_SERVICE);
40      View view = inflater.inflate(R.layout.preference_info, parent, false);
41      return view;
42  }
43
44  }
45
46
47
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\InterfaceBuscaGeralPorString.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4  * Created by Jucemar on 15/04/2016.
5  */
6 public interface InterfaceBuscaGeralPorString {
7     public void informarTermoParaConsulta(int tipoBusca,String termo);
8 }
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juccemar_dimon\cadesaude\InterfaceFiltroDeEstabelecimentos.java

```
1 package br.com.juccemar_dimon.cadesaude;
2
3 import java.util.ArrayList;
4
5 /**
6  * Created by Juccemar on 04/04/2016.
7  */
8 public interface InterfaceFiltroDeEstabelecimentos {
9     public static final int FILTRO_TIPO_ESTABELECIMENTO = 1;
10    public static final int FILTRO_TIPO_ATENDIMENTO = 2;
11    public static final int FILTRO_TIPO_CONVENIO = 3;
12    public static final int FILTRO_TIPO_SERVICO = 4;
13    public static final int FILTRO_TIPO_PROFISSIONAL = 5;
14
15    public void filtrarPorTipoDeEstabelecimento(ArrayList<String> tiposEstabelecimentosSelecionados);
16
17    public void filtrarPorTipoDeAtendimento(ArrayList<String> tiposAtendimentosSelecionados);
18
19    public void filtrarPorTipoDeConvenio(ArrayList<String> tiposConveniosSelecionados);
20
21    public void filtrarPorTipoDeProfissional(ArrayList<String> tiposProfissionaisSelecionados);
22
23    public void filtrarPorTipoDeServico(ArrayList<String> tiposServicosSelecionados);
24
25    public void limparTodosFiltros();
26 }
27
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\MainActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3  * Created by Jucemar on 04/04/2016.
4  */
5
6 import android.app.SearchManager;
7 import android.content.Context;
8 import android.content.Intent;
9 import android.content.SharedPreferences;
10 import android.os.AsyncTask;
11 import android.os.Bundle;
12 import android.preference.PreferenceManager;
13 import android.support.design.widget.CoordinatorLayout;
14 import android.support.design.widget.Snackbar;
15 import android.support.v4.view.MenuItemCompat;
16 import android.support.v7.app.AppCompatActivity;
17 import android.support.v7.widget.DefaultItemAnimator;
18 import android.support.v7.widget.LinearLayoutManager;
19 import android.support.v7.widget.RecyclerView;
20 import android.support.v7.widget.SearchView;
21 import android.support.v7.widget.Toolbar;
22 import android.util.Log;
23 import android.view.Menu;
24 import android.view.MenuInflater;
25 import android.view.MenuItem;
26 import android.view.View;
27 import android.widget.ImageView;
28 import android.widget.ProgressBar;
29 import android.widget.TextView;
30 import com.github.clans.fab.FloatingActionButton;
31
32 import java.util.ArrayList;
33 import java.util.HashSet;
34 import java.util.Set;
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\MainActivity.java

```
36 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
37     private SharedPreferences config;
38     private SharedPreferences.Editor editorConfig;
39     private Set<String> listaDeMunicipios;
40     private String dadosDeBuscaNaTela; RecyclerView
41     private meuRecyclerView; RecyclerView.LayoutManager
42     private meuLayoutManager; EstabelecimentoAdapter
43     private mAdapter;
44     private Set<String> listaDeFavoritos; fabBuscaGPS;
45     private FloatingActionButton fabBuscaGeral;
46     private FloatingActionButton EstabelecimentoDAO
47     private EstabelecimentoDAO estabelecimentos;
48     private ArrayList<Estabelecimento> bkpEstabelecimentos;
49     private ArrayList<Estabelecimento> barraDeProgresso;
50     private ProgressBar background;
51     private ImageView labelSemFavoritos;
52     private TextView posicaoNaLista;
53     private int CoordinatorLayout
54     private void executarBuscaPorPalavra(Intent intent) {
55         if (Intent.ACTION_SEARCH.equals(intent.getAction())) {
56             dadosDeBuscaNaTela = intent.getStringExtra(SearchManager.QUERY);
57         }
58     }
59
60     @Override
61     protected void onCreate(Bundle savedInstanceState) {
62         super.onCreate(savedInstanceState);
63         estabelecimentos = new ArrayList<>();
64         dadosDeBuscaNaTela = "";
65         posicaoNaLista = -1;
66         setContentView(R.layout.activity_main);
67         if (savedInstanceState != null) {
68             --
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\MainActivity.java

```
71     posicaoNaLista = savedInstanceState.getInt("posicaoNaLista");
72 }
73 config = getSharedPreferences("favoritos", MODE_PRIVATE);
74 layout = (CoordinatorLayout) findViewById(R.id.coordinator_layout_main_activity);
75 barraDeProgresso = (ProgressBar) findViewById(R.id.barra_progresso_main_activity);
76 background = (ImageView) findViewById(R.id.background_lista_vazia);
77 labelSemFavoritos = (TextView) findViewById(R.id.label_lista_sem_favoritos);
78 Log.e("BarraDeProgresso", (barraDeProgresso == null ? "é Null" : "não é null").toString());
79 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_main);
80 setSupportActionBar(toolbar);
81 meuRecyclerView = (RecyclerView) findViewById(R.id.lista_estabelecimentos);
82 meuRecyclerView.setHasFixedSize(true); meuLayoutManager = new
83     LinearLayoutManager(this);
84 meuRecyclerView.setLayoutManager(meuLayoutManager);
85 meuRecyclerView.setItemAnimator(new DefaultItemAnimator());
86 meuRecyclerView.scrollToPosition(posicaoNaLista);
87 meuRecyclerView.addOnScrollListener(new RecyclerView.OnScrollListener() {
88     @Override
89     public void onScrollStateChanged(RecyclerView recyclerView, int newState) {
90         super.onScrollStateChanged(recyclerView, newState);
91     }
92 }
93 @Override
94 public void onScrolled(RecyclerView recyclerView, int dx, int dy) {
95     super.onScrolled(recyclerView, dx, dy);
96     if (dy > 0) {
97         fabBuscaGPS.hide(true);
98         fabBuscaGeral.hide(true);
99     } else {
100         fabBuscaGPS.show(true);
101         fabBuscaGeral.show(true);
102     }
103 }
104 fabBuscaGPS = (FloatingActionButton) findViewById(R.id.fab_busca_gps);
105 ...
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\MainActivity.java

```
106 fabBuscaGPS.setOnClickListener(this);
107 fabBuscaGeral = (FloatingActionButton) findViewById(R.id.fab_busca_parametros);
108 fabBuscaGeral.setOnClickListener(this);
109 executarBuscaPorPalavra(getIntent());
110 }
111
112 @Override
113 protected void onSaveInstanceState(Bundle outState) {
114     super.onSaveInstanceState(outState);
115     outState.putInt("posicaoNaLista", posicaoNaLista);
116 }
117
118 public void checarFavoritos() {
119     if (estabelecimentos.size() == 0) {
120         labelSemFavoritos.setVisibility(View.VISIBLE);
121         background.setVisibility(View.VISIBLE);
122     }
123 }
124
125 private void carregaDados() {
126     CarregadorDeDados carregadorDeDados = new CarregadorDeDados(this);
127     carregadorDeDados.execute();
128 }
129
130 @Override
131 protected void onResume() {
132     super.onResume();
133     carregaDados();
134 }
135
136 @Override
137 protected void onNewIntent(Intent intent) {
138     super.onNewIntent(intent);
139     executarBuscaPorPalavra(intent);
140     ...
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\MainActivity.java

```
141     }
142
143     @Override
144     public boolean onCreateOptionsMenu(Menu menu) {
145         MenuInflater inflater = getMenuInflater();
146         inflater.inflate(R.menu.menu_main, menu);
147         MenuItem searchItem = menu.findItem(R.id.action_search);
148         SearchManager searchManager = (SearchManager) MainActivity.this
149             .getSystemService(Context.SEARCH_SERVICE);
150         SearchView searchView = null;
151         if (searchItem != null) {
152             searchView = (SearchView) searchItem.getActionView();
153         }
154         if (searchView != null) {
155             searchView.setSearchableInfo(searchManager.getSearchableInfo(MainActivity.this
156                 .getComponentName()));
157         }
158         MenuItemCompat.setOnActionExpandListener(searchView, new MenuItemCompat
159             .OnActionExpandListener() {
160             @Override
161             public boolean onMenuItemActionCollapse(MenuItem item) {
162                 carregaDados();
163                 return true;
164             }
165         });
166         @Override
167         public boolean onMenuItemActionExpand(MenuItem item) {
168             // Fazer algo quando o menu é expandido
169             return true;
170         }
171     };
172     MenuItemCompat.setOnActionExpandListener(searchItem, expandListener);
173     return super.onCreateOptionsMenu(menu);
174 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\MainActivity.java

```
176  @Override
177  public boolean onOptionsItemSelected(MenuItem item) {
178      int id = item.getItemId();
179      switch (id) {
180          case R.id.action_configuracoes:
181              Intent c = new Intent(this, PreferenciasActivity.class);
182              startActivity(c);
183              return true;
184          case R.id.action_ajuda:
185              Intent a = new Intent(this, AjudaActivity.class);
186              startActivity(a);
187              return true;
188          case R.id.action_sobre:
189              Intent s = new Intent(this, SobreActivity.class);
190              startActivity(s);
191              return true;
192          }
193      return super.onOptionsItemSelected(item);
194  }
195
196  @Override
197  public void onClick(View v) {
198      int id = v.getId();
199      switch (id) {
200          case R.id.fab_busca_gps:
201              Intent bgps = new Intent(this, BuscaGPSActivity.class);
202              bgps.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
203              startActivity(bgps);
204              break;
205          case R.id.fab_busca_parametros:
206              Intent bg = new Intent(this, BuscaGeraIActivity.class);
207              startActivity(bg);
208              break;
209      }
210  }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\MainActivity.java

```
211
212
213     protected void onPause() {
214         super.onPause();
215         if (meuRecyclerView != null) {
216             LinearLayoutManager lm = (LinearLayoutManager) meuRecyclerView.getLayoutManager();
217             posicaoNaLista = lm.findFirstVisibleItemPosition();
218         }
219     }
220
221     private class CarregadorDeDados extends AsyncTask<Void, Void, Void> {
222         private Context contexto;
223
224         public CarregadorDeDados(Context contexto) {
225             this.contexto = contexto;
226         }
227
228         @Override
229         protected Void doInBackground(Void... params) {
230             listaDeFavoritos = config.getStringSet("listaDeFavoritos", new HashSet<String>());
231             listaDeMunicipios = PreferenceManager.getDefaultSharedPreferences(contexto)
232                 .getStringSet(getString(R.string.pref_municipios_preferidos_key), new HashSet<String>());
233             estabDAO = EstabelecimentoDAO.getInstance(getApplicationContext());
234             estabelecimentos = estabDAO.listaEstabelecimentosFavoritos(listaDeFavoritos);
235             if (dadosDeBuscaNaTela.trim().length() > 0) {
236                 dadosDeBuscaNaTela = StringUtils.removeAcentos(dadosDeBuscaNaTela);
237                 dadosDeBuscaNaTela = dadosDeBuscaNaTela.trim();
238                 bkpEstabelecimentos = new ArrayList<>();
239                 bkpEstabelecimentos = estabelecimentos;
240                 ArrayList<Estabelecimento> estabelecimentosTemp = new ArrayList<Estabelecimento>();
241                 for (Estabelecimento e : estabelecimentos) {
242                     if (e.getNomeFantasia().toLowerCase().contains(dadosDeBuscaNaTela.toLowerCase()) || e.
243                         getRazaoSocial().toLowerCase().contains(dadosDeBuscaNaTela)
244                             .contains(dadosDeBuscaNaTela)) {
245                         estabelecimentosTemp.add(e);
246                     }
247                 }
248             }
249             estabelecimentos = estabelecimentosTemp;
250         }
251     }
252 }
```


File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\MainActivity.java

```
245     }
246     }
247     estabelecimentos = estabelecimentosTemp;
248 }
249 return null;
250 }
251
252 @Override
253 protected void onPostExecute(Void aVoid) {
254     super.onPostExecute(aVoid);
255     mAdapter = new EstabelecimentoAdapter(contexto, estabelecimentos, listaDeFavoritos);
256     mRecyclerView.setAdapter(mAdapter);
257     if (estabelecimentos.size() > 0) {
258         mRecyclerView.setVisibility(View.VISIBLE);
259         barraDeProgresso.setVisibility(View.GONE);
260         background.setVisibility(View.GONE);
261         labelSemFavoritos.setVisibility(View.GONE);
262     } else {
263         mRecyclerView.setVisibility(View.GONE);
264         barraDeProgresso.setVisibility(View.GONE);
265         background.setVisibility(View.VISIBLE);
266         labelSemFavoritos.setVisibility(View.VISIBLE);
267     }
268     if (posicaoNaLista > -1) {
269         mRecyclerView.scrollToPosition(posicaoNaLista);
270     }
271     Log.e("Inicio", "Lista de municipio " + listaDeMunicipios.size());
272     if (listaDeMunicipios.size() == 0) {
273         Intent i = new Intent(contexto, PrimeirosPassosActivity.class);
274         startActivity(i);
275     }
276     if (dadosDeBuscaNaTela.length() > 0) {
277         Snackbar.make(layout, "Foram localizados " + mAdapter.getItemCount() + " estabelecimentos", Snackbar.
278             LENGTH_SHORT).show();
279     }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\MainActivity.java

```
279     dadosDeBuscaNaTela = "";  
280 }  
281  
282 @Override  
283     protected void onCreate(Bundle savedInstanceState) {  
284     super.onCreate(savedInstanceState);  
285     setContentView(R.layout.activity_main);  
286     // Inicializa o RecyclerView  
287     RecyclerView recyclerView = findViewById(R.id.recycler_view);  
288     recyclerView.setLayoutManager(new LinearLayoutManager(this));  
289     // Carrega os dados da base de dados  
290     CarregarDados();  
291 }  
292
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juccemar_dimon\cadesaude\PreferenciasActivity.java

```
1 package br.com.juccemar_dimon.cadesaude;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.view.MenuItem;
6
7 /**
8  * Created by Juccemar on 04/04/2016.
9  */
10 public class PreferenciasActivity extends AppCompatActivity {
11
12     private PreferenciasFragment preferenciasFragment;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         preferenciasFragment = new PreferenciasFragment();
18         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
19         if (savedInstanceState == null) {
20             getSupportFragmentManager().beginTransaction()
21                 .replace(android.R.id.content, preferenciasFragment, "preferenciasFragment")
22                 .commit();
23         }
24     }
25
26     @Override
27     public boolean onOptionsItemSelected(MenuItem item) {
28         int id = item.getItemId();
29         switch (id) {
30             case android.R.id.home:
31                 onBackPressed();
32             }
33         return super.onOptionsItemSelected(item);
34     }
35 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PreferencesActivity.java

36 }

37

38

39

40

41

42

43

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PreferencesFragment.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.SharedPreferences;
4 import android.os.Bundle;
5 import android.preference.Preference.MultiSelectListPreference;
6 import android.preference.PreferenceFragment;
7 import android.preference.PreferenceManager;
8 import android.support.design.widget.Snackbar;
9 import android.util.Log;
10 import android.view.View;
11
12 import java.util.ArrayList;
13 import java.util.Arrays;
14 import java.util.HashSet;
15 import java.util.Set;
16
17 /**
18  * Created by Jucemar on 14/02/2016.
19  */
20 public class PreferencesFragment extends PreferenceFragment implements SharedPreferences.
    OnSharedPreferenceChangeListener {
21
22     private MultiSelectListPreference listaDePreferencias;
23     private Set<String> selecoesNaLista;
24     private SharedPreferences preferencias;
25     private ArrayList<String> codMunicipio;
26     private ArrayList<String> nomeMunicipio;
27     private String municipioPadrao;
28     private SharedPreferences.Editor editorConfig;
29     private SeekBarPreference mSeek2;
30     private InfoPreference info;
31
32     @Override
33     public void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PreferenciasFragment.java

```
35     addPreferencesFromResource(R.xml.preferences);
36     listaDePreferencias = (MultiSelectListPreference) findPreference(getString(R.string.
    pref_municipios_preferidos_key));
37     preferencias = PreferenceManager.getDefaultSharedPreferences(getActivity());
38     selecoesNaLista = new HashSet<>();
39     info = (InfoPreference) findPreference(getString(R.string.pref_info_key));
40     getPreferenceScreen().removePreference(info);
41     if (getArguments() != null) {
42         municipioPadrao = getArguments().getString("municipioAtual");
43         Log.e("municipioAtual",
            municipioPadrao);
44         selecoesNaLista.add(municipioPadrao);
45         listaDePreferencias.setValues(selecoesNaLista);
46         editorConfig = listaDePreferencias.getEditor();
47         editorConfig.putStringSet("municipios_preferidos", selecoesNaLista);
48         editorConfig.apply();
49         getPreferenceScreen().addPreference(info);
50     }
51     mSeek2 = (SeekBarPreference) findPreference(getString(R.string.pref_distancia_estabelecimentos_key));
52     String[] arrayNomes = getResources().getStringArray(R.array.nomes_municipios_view);
53     nomeMunicipio = new ArrayList<String>(Arrays.asList(arrayNomes));
54     String[] arrayCodigos = getResources().getStringArray(R.array.codigos_municipios);
55     codMunicipio = new ArrayList<String>(Arrays.asList(arrayCodigos));
56     mostrarMunicipiosEscolhidos(listaDePreferencias.getValues());
57 }
58
59 @Override
60 public void onViewCreated(View view, Bundle savedInstanceState) {
61     super.onViewCreated(view, savedInstanceState);
62     if (getArguments() != null) {
63         Snackbar.make(view, "Você está em " + listaDePreferencias.getSummary(), Snackbar.LENGTH_SHORT).show();
64     }
65 }
66
67 @Override
68 public void onResume() {
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PreferenciasFragment.java

```
69     super.onResume();
70     SharedPreferences prefs = getSharedPreferences("manager").getSharedPreferences();
71     onSharedPreferencesChanged(prefs, getString(R.string.pref_distancia_estabelecimentos_key));
72     prefs.registerOnSharedPreferenceChangeListener(this);
73 }
74
75
76 @Override
77 public void onPause() {
78     super.onPause();
79     SharedPreferences prefs = getSharedPreferences("manager").getSharedPreferences();
80     prefs.unregisterOnSharedPreferenceChangeListener(this);
81 }
82
83
84 @Override
85 public void onSharedPreferenceChanged(SharedPreferences prefs, String key) {
86
87     if (key.equals(getString(R.string.pref_municipios_preferidos_key))) {
88         StringBuffer sb = new StringBuffer();
89         Set<String> temp = prefs.getStringSet(key, new HashSet<String>());
90         if (temp.size() > 0) {
91             getActivity().setResult(1);
92         }
93         ArrayList<String> tempList = new ArrayList(temp);
94         for (int i = 0; i < tempList.size(); i++) {
95             int j = codMunicipio.indexOf(tempList.get(i));
96             sb.append(nomeMunicipio.get(j));
97             if (i < tempList.size() - 1) {
98                 sb.append(", ");
99             }
100             sb.append(". ");
101
102             listaDePreferencias.setSummary(sb.toString());
103         }
104     }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PreferencesFragment.java

```
104     Log.e("teste", "Chamou o onSharedePrefer  ce");
105 }
106 if (key.equals(getString(R.string.pref_distancia_estabelecimentos_key))) {
107     int i = prefs.getInt(key, 1);
108     mSeek2.setSummary(i + " Km");
109 }
110 }
111
112 private void mostrarMunicipiosEscolhidos(Set<String> setMunicipios) {
113     StringBuffer sb = new StringBuffer(); Set<String>
114     temp = setMunicipios; ArrayList<String> tempList =
115     new ArrayList(temp);
116     for (int i = 0; i < tempList.size(); i++) {
117         int j = codMunicipio.indexOf(tempList.get(i));
118         sb.append(nomeMunicipio.get(j));
119         if (i < tempList.size() - 1) {
120             sb.append(", ");
121         }
122     }
123     sb.append(". ");
124     if (setMunicipios.size() > 0) {
125         listaDePreferencias.setSummary(sb.toString());
126     } else {
127         listaDePreferencias.setSummary("  o h  munic pios selecionados.");
128     }
129 }
130 }
131 }
132 }
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3  * Created by Jucemar on 14/02/2016.
4  */
5
6 import android.Manifest;
7 import android.app.Activity;
8 import android.app.AlertDialog;
9 import android.content.Context;
10 import android.content.Intent;
11 import android.content.IntentSender;
12 import android.content.SharedPreferences;
13 import android.content.pm.PackageManager;
14 import android.graphics.Color;
15 import android.location.Location;
16 import android.os.AsyncTask;
17 import android.os.Bundle;
18 import android.support.design.widget.Snackbar;
19 import android.support.v4.app.ActivityCompat;
20 import android.support.v7.app.AppCompatActivity;
21 import android.util.Log;
22 import android.view.Menu;
23 import android.view.MenuInflater;
24 import android.view.MenuItem;
25 import android.view.View;
26 import android.widget.FrameLayout;
27 import android.widget.ImageView;
28 import android.widget.ProgressBar;
29 import android.widget.TextView;
30 import com.google.android.gms.common.ConnectionResult;
31 import com.google.android.gms.common.GoogleApiAvailability;
32 import com.google.android.gms.common.api.GoogleApiClient;
33 import com.google.android.gms.common.api.GoogleApiClient.ConnectionCallbacks;
34 import com.google.android.gms.common.api.PendingResult;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
36 import com.google.android.gms.common.api.ResultCallback;
37 import com.google.android.gms.common.api.Status;
38 import com.google.android.gms.location.LocationListener;
39 import com.google.android.gms.location.LocationRequest;
40 import com.google.android.gms.location.LocationServices;
41 import com.google.android.gms.location.LocationSettingsRequest;
42 import com.google.android.gms.location.LocationSettingsResult;
43 import com.google.android.gms.location.LocationSettingsStates;
44 import com.google.android.gms.location.LocationSettingsStatusCodes;
45
46
47 public class PrimeiraConfiguracaoActivity extends AppCompatActivity implements
48     CaixaDialogoGoogleAPI.ErroGooglePlayService,
49     ConnectionCallbacks,
50     GoogleApiClient.OnConnectionFailedListener,
51     LocationListener, ResultCallback<LocationSettingsResult> {
52
53     public static final int REQUEST_CODE_GOOGLE_PLAY_SERVICES = 1982;
54     private static final int REQUEST_CODE_LOCATION = 2;
55     private LocationRequest request;
56     private GoogleApiClient apiClient;
57     private SharedPreferences config;
58     private Status status;
59     private Location localizacao; CarregadorDeDados
60     private carregadorDeDados; PreferenciasFragment
61     private preferenciasFragment; ProgressBar
62     private barraDeProgresso;
63     private TextView txtErroGps;
64     private ImageView imgErroGps;
65     private FrameLayout layoutActivity;
66     private MenuItem botaoConfirmar;
67
68     @Override
69     protected void onCreate(Bundle savedInstanceState) {
70         super.onCreate(savedInstanceState);
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_\dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
71 setContentView(R.layout.activity_primeira_configuracao);
72 preferenciasFragment = new PreferenciasFragment();
73 txtErroGps = (TextView) findViewById(R.id.label_erro_primeira_config_activity);
74 imgErroGps = (ImageView) findViewById(R.id.background_erro_primeira_config_activity);
75 barraDeProgresso = (ProgressBar) findViewById(R.id.barra_progresso_primeira_config_activity);
76 layoutActivity = (FrameLayout) findViewById(R.id.layout_primeira_config_activity);
77 //metodo chamado para para realizar verificações para saber se o dispositivo possui o google play services
    instalado
78 verificarGooglePlayServiceO;
79 //Instancia o objeto da api do google e apartir de dai realiza checagens quanto ao google play services
80 apiClient = new GoogleApiClient.Builder(this)
81     .addConnectionCallbacks(this)
82     .addOnConnectionFailedListener(this)
83     .addApi(LocationServices.API)
84     .build();
85 }
86
87 private void carregarDados() {
88     carregadorDeDados = new CarregadorDeDados(this);
89     carregadorDeDados.execute();
90 }
91
92 @Override
93 protected void onSaveInstanceState(Bundle outState) {
94     super.onSaveInstanceState(outState);
95 }
96
97 private void verificarGooglePlayService() {
98     GoogleApiAvailability api = GoogleApiAvailability.getInstance();
99     int code = api.isGooglePlayServicesAvailable(getApplicationContext());
100     if (code == ConnectionResult.SUCCESS) {
101         //da sequencia nos procedimento normalmente
102     } else {
103         if (code == ConnectionResult.SERVICE_MISSING ||
104             code == ConnectionResult.SERVICE_VERSION_UPDATE_REQUIRED ||
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
105         code == ConnectionResult.SERVICE_DISABLED) {
106             Dialog dialog = api.getErrorDialog(this, code, REQUEST_CODE_GOOGLE_PLAY_SERVICES);
107             dialog.show();
108         } else {
109             CaixaDeDialogoGoogIeAPI cxDialog = new CaixaDeDialogoGoogIeAPI();
110             cxDialog.setErrorGoogIePlayService(this);
111             cxDialog.show(getFragmentManager(), "DialogErro");
112         }
113     }
114 }
115
116 @Override
117 protected void onStart() {
118     super.onStart();
119     apiClient.connect();
120 }
121
122 @Override
123 public void onBackPressed() {
124     super.onBackPressed();
125 }
126
127 @Override
128 protected void onStop() {
129     super.onStop();
130     apiClient.disconnect();
131 }
132
133 @Override
134 protected void onPause() {
135     super.onPause();
136 }
137
138 @Override
139 public void onConnected(Bundle bundle) {
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
140     request = LocationRequest.createO;
141     request.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
142     request.setInterval(500);
143     LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder().addLocationRequest(request);
144     builder.setAlwaysShow(true);
145     PendingResult<LocationSettingsResult> result = LocationServices.SettingsApi.checkLocationSettings(apiClient,
        builder.build());
146     result.setResultCallback(this);
147 }
148
149 @Override
150 public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
151     if (requestCode == REQUEST_CODE_LOCATION) {
152         if (grantResults.length == 1 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
153             adquirirLocalizacao();
154         } else {
155             Snackbar.make(layoutActivity, R.string.tentar_novamente_label, Snackbar.LENGTH_INDEFINITE)
156                 .setAction(R.string.tentar_novamente_action, new View.OnClickListener() {
157                     @Override
158                     public void onClick(View v) {
159                         adquirirLocalizacao();
160                     }
161                 })
162                 .setActionTextColor(Color.YELLOW)
163                 .show();
164             imgErroGps.setVisibility(View.VISIBLE);
165             txtErroGps.setVisibility(View.VISIBLE);
166             barraDeProgresso.setVisibility(View.GONE);
167         }
168     }
169
170     private void adquirirLocalizacao() {
171         imgErroGps.setVisibility(View.GONE);
172         txtErroGps.setVisibility(View.GONE);
173     }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
174 barraDeProgresso.setVisibility(View.VISIBLE);
175 if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.
    PERMISSION_GRANTED) {
176     if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.ACCESS_FINE_LOCATION)) {
177         } else {
178             ActivityCompat.requestPermissions(this, new String[] {Manifest.permission.ACCESS_FINE_LOCATION},
    REQUEST_CODE_LOCATION);
179         }
180     } else {
181         LocationServices.FusedLocationApi.requestLocationUpdates(apiClient, request, this);
182     }
183 }
184
185 @Override
186 public void onConnectionSuspended(int i) {
187     //Não implementado
188 }
189
190 @Override
191 public void onConnectionFailed(ConnectionResult connectionResult) {
192     //Não implementado
193 }
194
195 @Override
196 public boolean onCreateOptionsMenu(Menu menu) {
197     getMenuInflater = getMenuInflater();
198     getMenuInflater.inflate(R.menu.menu_primeira_configuracao, menu);
199     botaoConfirmar = menu.findItem(R.id.action_salvar_config);
200     botaoConfirmar.setVisible(false);
201     return super.onCreateOptionsMenu(menu);
202 }
203
204 @Override
205 public boolean onOptionsItemSelected(MenuItem item) {
206     int id = item.getItemId();
207     ~~~~~
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
207     switch (id) {
208         case R.id.action_salvar_config:
209             Intent i = new Intent(PrimeiraConfiguracaoActivity.this, MainActivity.class);
210             i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);
211             startActivity(i);
212             finish();
213         }
214         return super.onOptionsItemSelected(item);
215     }
216
217     @Override
218     public void onLocationChanged(Location location) { this.localizacao = location;
219         LocationServices.FusedLocationApi.removeLocationUpdates(apiClient, this);
220         carregarDados();
221     }
222
223
224     @Override
225     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
226         final LocationSettingsStates states;
227         if (data != null) {
228             states = LocationSettingsStates.fromIntent(data);
229         }
230         switch (requestCode) {
231             case REQUEST_CODE_GOOGLE_PLAY_SERVICES:
232                 switch (resultCode) {
233                     case Activity.RESULT_OK:
234                         adquirirLocalizacao();
235                         break;
236                     case Activity.RESULT_CANCELED:
237                         Snackbar.make(layoutActivity, R.string.tentar_novamente_label, Snackbar.LENGTH_INDEFINITE)
238                             .setAction(R.string.tentar_novamente_action, new View.OnClickListener() {
239                                 @Override
240                                     public void onClick(View v) {
241                                         try {
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
242         status.startResolutionForResult(PrimeiraConfiguracaoActivity.this,
243             REQUEST_CODE_GOOGLE_PLAY_SERVICES);
244     } catch (IntentSender.SendIntentException e) {
245         e.printStackTrace();
246     }
247 }
248 .setActionTextColor(Color.YELLOW)
249 .show();
250 imgErroGps.setVisibility(View.VISIBLE);
251 txtErroGps.setVisibility(View.VISIBLE);
252 barraDeProgresso.setVisibility(View.GONE);
253 }
254 break;
255 }
256 }
257
258 @Override
259 public void onActivityResult(LocationSettingsResult locationSettingsResult) {
260     status = locationSettingsResult.getStatus();
261     switch (status.getStatusCode()) {
262         case LocationSettingsStatusCodes.SUCCESS:
263             adquirirLocalizacao();
264             break;
265         case LocationSettingsStatusCodes.RESOLUTION_REQUIRED:
266             try {
267                 status.startResolutionForResult(PrimeiraConfiguracaoActivity.this,
268                     REQUEST_CODE_GOOGLE_PLAY_SERVICES);
269             } catch (IntentSender.SendIntentException e) {
270                 Snackbar.make(layoutActivity, "Algun motivo desconhecido impediu a aquisição de sua localização",
271                     Snackbar.LENGTH_LONG).setActionTextColor(Color.RED).show();
272                 imgErroGps.setVisibility(View.VISIBLE);
273                 txtErroGps.setVisibility(View.VISIBLE);
274                 barraDeProgresso.setVisibility(View.GONE);
275             }
276         }
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
274         break;
275         case LocationSettingsStatusCodes.SETTINGS_CHANGE_UNAVAILABLE:
276             Snackbar.make(layoutActivity, "Algun motivo desconhecido impediu a aquisição de sua localização",
277                 Snackbar.LENGTH_LONG).setActionTextColor(Color.RED).show();
278             imgErroGps.setVisibility(View.VISIBLE);
279             txtErroGps.setVisibility(View.VISIBLE);
280             barraDeProgresso.setVisibility(View.GONE);
281             break;
282         }
283     }
284     @Override
285     protected void onResume() {
286         super.onResume();
287     }
288
289
290     private void abrirPreferencias(String resultado) {
291         Bundle b = new Bundle();
292         b.putString("municipioAtual", resultado);
293         preferenciasFragment.setArguments(b);
294         fragmentManager.beginTransaction()
295             .replace(android.R.id.content, preferenciasFragment, "preferenciasFragment")
296             .commit();
297         botaoConfirmar.setVisible(true);
298     }
299
300     @Override
301     public void googlePlayServiceIndisponivel() {
302         onBackPressed();
303         finish();
304     }
305
306     private class CarregadorDeDados extends AsyncTask<Void, Void, String> {
307         private Context contexto;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PrimeiraConfiguracaoActivity.java

```
308 private EstabelecimentoDAO estabDAO;
309 private String resultado = "";
310
311 public CarregadorDeDados(Context contexto) {
312     this.contexto = contexto;
313 }
314
315 @Override
316 protected void onPreExecute() {
317     super.onPreExecute();
318     imgErroGps.setVisibility(View.GONE);
319     txtErroGps.setVisibility(View.GONE);
320     barraDeProgresso.setVisibility(View.VISIBLE);
321 }
322
323 @Override
324 protected String doInBackground(Void... params) {
325     Log.e("Estabelecimentos", "Carregando dados em background");
326     estabDAO = EstabelecimentoDAO.getInstance(getApplicationContext());
327     String resultado = estabDAO.buscaCodigoMunicipioAtual(localizacao);
328     Log.e("resultadoMunicipio", resultado);
329     return resultado;
330 }
331
332 @Override
333 protected void onPostExecute(String resultado) {
334     super.onPostExecute(resultado);
335     imgErroGps.setVisibility(View.GONE);
336     txtErroGps.setVisibility(View.GONE);
337     barraDeProgresso.setVisibility(View.GONE);
338     abrirPreferencias(resultado);
339 }
340
341 ...
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PrimeirosPassosActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.support.v7.app.AppCompatActivity;
6 import android.view.View;
7 import android.widget.Button;
8
9 /**
10  * Created by Jucemar on 14/02/2016.
11  */
12 public class PrimeirosPassosActivity extends AppCompatActivity {
13
14     private Button botaoConfig;
15     private static final int CONFIGURACAO_MUNICIPIOS = 1;
16
17     private final View.OnClickListener mDelayHideTouchListener = new View.OnClickListener() {
18         @Override
19         public void onClick(View v) {
20             Intent i = new Intent(PrimeirosPassosActivity.this, PrimeiraConfiguracaoActivity.class);
21             startActivity(i);
22             finish();
23         }
24     };
25
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         setContentView(R.layout.activity_primeiros_passos);
30         botaoConfig = (Button) findViewById(R.id.dummy_button);
31         botaoConfig.setOnClickListener(mDelayHideTouchListener);
32     }
33
34     @Override
35     public void onBackPressed() {
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\PrimeirosPassosActivity.java

```
36     SairDialog s = new SairDialog();
37     s.show(getFragmentManager, "Sairdialog");
38 }
39
40 }
41
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ProfissionaisDAO.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.util.Log;
7
8 import java.util.ArrayList;
9 import java.util.Set;
10
11 /**
12  * Created by Jucemar on 10/03/2016.
13  */
14 public class ProfissionaisDAO {
15
16     private static ProfissionaisDAO instancia;
17     private SQLiteDatabase bd;
18
19     public static ProfissionaisDAO getInstancia(Context contexto) {
20         if (instancia == null) {
21             instancia = new ProfissionaisDAO(contexto);
22         }
23         return instancia;
24     }
25
26     private ProfissionaisDAO(Context contexto) {
27         BancoDeDados bancoDeDados = BancoDeDados.getInstancia(contexto);
28         bd = bancoDeDados.getWritableDatabase();
29     }
30
31     public CharSequence[] listarTiposProfissionais(Set<String> setMunicipios){
32         CharSequence[] resultado=null;
33
34         if(setMunicipios.size()>0) {
35             -
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar\dimon\cadesaude\ProfissionaisDAO.java

```
36
37
38     StringBuffer filtro = new StringBuffer();
39     String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
40
41
42
43
44     for (int i = 0; i < municipios.length; i++) {
45         filtro.append("estabelecimento."+TabelaEstabelecimento.Colunas.CODIGO_MUNICIPIO + "=? ");
46         if (i < municipios.length - 1) {
47             filtro.append("OR ");
48         }
49     }
50
51     String query = "SELECT distinct profissional.cbo_descricao FROM estabelecimento JOIN
52     vinculo_profissional_estabelecimento ON cnes=vinculo_profissional_estabelecimento.estabelecimento_cnes" +
53     " JOIN profissional ON profissional.id=vinculo_profissional_estabelecimento.profissional_id WHERE " +
54     +filtro+"order by profissional.cbo_descricao;";
55     Log.e("query",query);
56     Cursor c = bd.rawQuery(query,municipios);
57     resultado=new CharSequence[c.getCount()];
58     int i=0;
59     if (c.moveToFirst()) {
60         do {
61             resultado[i]=c.getString(c.getColumnIndex(TabelaProfissionais.Colunas.CBO_DESCRICA0));
62             i++;
63         } while (c.moveToNext());
64     }
65     c.close();
66
67
68 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar\dimon\cadesaude\ProfissionaisDAO.java

```
69
70
71     return resultado;
72
73 }
74
75 public ArrayList<Estabelecimento> buscaStringEmNomeProfissionais(Set<String> setMunicipios, String sentenca){
76     StringBuffer filtroMunicipios = new StringBuffer();
77     String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
78
79     for (int i = 0; i < municipios.length; i++) {
80         filtroMunicipios.append("'" + municipios[i] + "'");
81         if (i < municipios.length - 1) {
82             filtroMunicipios.append(", ");
83         }
84     }
85
86     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
87     String query="SELECT * FROM estabelecimento WHERE cnes IN (SELECT DISTINCT estabelecimento.cnes FROM
        profissional \"n\" +
88         \"JOIN vinculo_profissional_estabelecimento ON profissional.id=vinculo_profissional_estabelecimento.
        profissional_id \"n\" +
89         \"JOIN estabelecimento ON estabelecimento.cnes=vinculo_profissional_estabelecimento.
        estabelecimento_cnes WHERE profissional.nome LIKE '%" +sentenca+"%' AND estabelecimento.codigo_municipio IN ('+
        filtroMunicipios+"'))";
90
91     Log.e("query", query);
92     Cursor c = bd.rawQuery(query, null);
93
94     if (c.moveToFirst()) {
95
96         do {
97             Estabelecimento e = EstabelecimentoDAO.extrairDoCursor(c);
98             resultado.add(e);
99         } while (c.moveToNext());
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ProfissionaisDAO.java

```
100     }
101     c.close();
102
103     return resultado;
104
105 }
106
107 public ArrayList<Estabelecimento> buscaStringEmNomeEspecialidade(Set<String> setMunicipios, String sentenca){
108     StringBuffer filtroMunicipios = new StringBuffer();
109     String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
110
111     for (int i = 0; i < municipios.length; i++) {
112         filtroMunicipios.append("'" + municipios[i] + "'");
113         if (i < municipios.length - 1) {
114             filtroMunicipios.append(", ");
115         }
116     }
117
118     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
119     String query="SELECT * FROM estabelecimento WHERE cnes IN (SELECT DISTINCT estabelecimento.cnes FROM
120     profissional \n" +
121     "      " + filtroMunicipios + ")";
122
123     Log.e("query", query);
124     Cursor c = bd.rawQuery(query, null);
125
126     if (c.moveToFirst()) {
127         do {
128             Estabelecimento e = EstabelecimentoDAO.extrairDoCursor(c);
129             resultado.add(e);
130         }
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ProfissionaisDAO.java

```
131         } while (c.moveToNext());
132     }
133     c.close();
134
135     return resultado;
136
137 }
138
139 public ArrayList<Profissional> listaProfissionais(String cnes) {
140     String[] argumentos = {cnes};
141
142     String query = "SELECT * FROM vinculo_profissional_estabelecimento JOIN profissional ON
143     vinculo_profissional_estabelecimento.profissional_id=profissional.id WHERE vinculo_profissional_estabelecimento.
144     estabelecimento_cnes=?";
145     Cursor c = bd.rawQuery(query, argumentos);
146
147     Log.i("Cursor", String.valueOf(c.getCount()));
148     ArrayList<Profissional> resultado = new ArrayList<Profissional>();
149     //Cursor c = bd.query(TabelaProfissionais.NOME_TABELA, null, TabelaProfissionais.Colunas.ID + "=?",
150     idsProfissionais, null, null, TabelaProfissionais.Colunas.NOME);
151     if (c.moveToFirst() {
152         do {
153             Profissional P = extrairDoCursor(c);
154             resultado.add(P);
155         } while (c.moveToNext());
156     } c.close();
157     Log.e("ResultadobuscaProf", String.valueOf(resultado.size()));
158     return resultado;
159 }
160
161 private static Profissional extrairDoCursor(Cursor c) {
162     String nome = c.getString(c.getColumnIndex(TabelaProfissionais.Colunas.NOME));
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ProfissionaisDAO.java

```
163     String cbo = c.getString(c.getColumnIndex(TabelaProfissionais.COLUNAS.CBO));
164     String cboDescricao = c.getString(c.getColumnIndex(TabelaProfissionais.COLUNAS.CBO_DESCRICAO));
165
166     return new Profissional(nome, cbo, cboDescricao);
167 }
168 }
169
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ProfissionaisFragment.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4  * Created by Jucemar on 18/03/2016.
5  */
6
7 import android.os.Bundle;
8 import android.support.v4.app.Fragment;
9 import android.view.LayoutInflater;
10 import android.view.View;
11 import android.view.ViewGroup;
12 import android.widget.ListView;
13
14 import java.util.ArrayList;
15
16 public class ProfissionaisFragment extends Fragment {
17
18     public ProfissionaisFragment() {
19     }
20
21     @Override
22     public void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24     }
25
26     @Override
27     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
28         View v = inflater.inflate(R.layout.fragment_listas, container, false);
29         ListView lv = (ListView) v.findViewById(R.id.list_view);
30         lv.setDivider(null);
31         ArrayList<Profissional> listaDeProfissionais = getArguments().getParcelableArrayList("listaDeProfissionais");
32         if (listaDeProfissionais.size() > 0) {
33             ProfissionalAdapter pa = new ProfissionalAdapter(getActivity(), listaDeProfissionais);
34             lv.setAdapter(pa);
35         } else {
36             -
37         }
38     }
39 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ProfissionaisFragment.java

```
36     ArrayList<Profissional> erro = new ArrayList<Profissional>();
37     erro.add(new Profissional("Não há informações cadastradas", "Desculpe o transtorno"));
38     ProfissionalAdapter sa = new ProfissionalAdapter(getActivity(), erro);
39     lv.setAdapter(sa);
40 }
41     return v;
42 }
43
44 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Profissional.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.os.Parcel;
4 import android.os.Parcelable;
5
6 /**
7  * Created by Jucemar on 14/02/2016.
8  */
9 public class Profissional implements Parcelable {
10
11     private String nome;
12     private String cbo;
13     private String cboDescricao;
14
15     public Profissional() {
16     }
17
18     public Profissional(String nome, String cbo, String cboDescricao) {
19         this.nome = nome;
20         this.cbo = cbo;
21         this.cboDescricao = cboDescricao;
22     }
23
24     protected Profissional(Parcel in) {
25         nome = in.readString(); cbo =
26         in.readString(); cboDescricao =
27         in.readString();
28     }
29
30     public static final Creator<Profissional> CREATOR = new Creator<Profissional>() {
31         @Override
32         public Profissional createFromParcel(Parcel in) {
33             return new Profissional(in);
34         }
35     }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Profissional.java

```
36  @Override
37  public Profissional[] newArray(int size) {
38      return new Profissional[size];
39  }
40  };
41
42  public Profissional(String s, String sl) {
43      this.nome = s;
44      this.cboDescricao = sl;
45  }
46
47  public String getNome() {
48      return nome;
49  }
50
51  public void setNome(String nome) {
52      this.nome = nome;
53  }
54
55  public String getCbo() {
56      return cbo;
57  }
58
59  public void setCbo(String cbo) {
60      this.cbo = cbo;
61  }
62
63  public String getCboDescricao() {
64      return cboDescricao;
65  }
66
67  public void setCboDescricao(String cboDescricao) {
68      this.cboDescricao = cboDescricao;
69  }
69  --
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\Profissional.java

```
71  @Override
72  public int describeContents() {
73      return 0;
74  }
75
76  @Override
77  public void writeToParcel(Parcel dest, int flags) {
78      dest.writeString(nome);
79      dest.writeString(cbo);
80      dest.writeString(cboDescricao);
81  }
82
83  }
84
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ProfissionalAdapter.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.AdapterView;
8 import android.widget.AdapterView.OnItemClickListener;
9
10 import java.util.ArrayList;
11
12 /**
13  * Created by Jucemar on 19/03/2016.
14  */
15 public class ProfissionalAdapter extends ArrayAdapter<Profissional> {
16
17     public ProfissionalAdapter(Context context, ArrayList<Profissional> objects) {
18         super(context, R.layout.fragment_listas, objects);
19     }
20
21     @Override
22     public View getView(int position, View convertView, ViewGroup parent) {
23         Profissional p = getItem(position);
24         if (convertView == null) {
25             convertView = LayoutInflater.from(getContext()).inflate(R.layout.item_listas, parent, false);
26
27             TextView tvNome = (TextView) convertView.findViewById(R.id.item_linha_1);
28             TextView tvCargo = (TextView) convertView.findViewById(R.id.item_linha_2);
29             tvNome.setText(p.getNome());
30             tvCargo.setText(p.getDescricao());
31             return convertView;
32         }
33     }
34 }
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\SairDialog.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.app.Activity;
4 import android.app.AlertDialog;
5 import android.app.Dialog;
6 import android.app.DialogFragment;
7 import android.content.DialogInterface;
8 import android.os.Bundle;
9
10 /**
11  * Created by Jucemar on 18/08/2016.
12  */
13 public class SairDialog extends DialogFragment implements DialogInterface.OnClickListener {
14     private PrimeirosPassosActivity contexto;
15
16     @Override
17     public void onAttach(Activity activity) {
18         super.onAttach(activity);
19         this.contexto=(PrimeirosPassosActivity)activity;
20     }
21
22     public SairDialog() {
23
24     }
25
26     @Override
27     public void onClick(DialogInterface dialog, int which) {
28         switch (which) {
29             case DialogInterface.BUTTON_POSITIVE:
30                 dismiss();
31                 contexto.finishAffinity();
32                 System.exit(0);
33                 break;
34             case DialogInterface.BUTTON_NEGATIVE:
35                 dismiss();
36         }
37     }
38 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\SairDialog.java

```
36         break;
37     }
38 }
39
40 @Override
41 public Dialog onCreateDialog(Bundle savedInstanceState) { AlertDialog.Builder
42     builder = new AlertDialog.Builder(getActivity());
43     builder.setMessage("Confirma a saída do aplicativo antes de configurar?")
44         .setTitle("Confirmação de saída")
45         .setPositiveButton("Sair", this)
46         .setNegativeButton("Cancelar", this);
47     return builder.create();
48 }
49
50 }
51
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juccemar_dimon\cadesaude\SeekBarPreference.java

```
1 package br.com.juccemar_dimon.cadesaude;
2 /**
3  * Created by Juccemar on 04/05/2016.
4  */
5
6 import android.content.Context;
7 import android.preference.Preference;
8 import android.util.AttributeSet;
9 import android.view.LayoutInflater;
10 import android.view.View;
11 import android.view.ViewGroup;
12 import android.widget.SeekBar;
13 import android.widget.SeekBar.OnSeekBarChangeListener;
14
15 public class SeekBarPreference extends Preference implements OnSeekBarChangeListener {
16     private SeekBar mSeekBar;
17     private int mProgress;
18
19     public SeekBarPreference(Context context) {
20         super(context);
21     }
22
23     public SeekBarPreference(Context context, AttributeSet attrs) {
24         super(context, attrs);
25     }
26
27     public SeekBarPreference(Context context, AttributeSet attrs, int defStyleAttr) {
28         super(context, attrs, defStyleAttr);
29     }
30
31     @Override
32     protected View onCreateView(ViewGroup parent) {
33         super.onCreateView(parent);
34         LayoutInflater inflater = (LayoutInflater) getContext().getSystemService(Context.LAYOUT_INFLATER_SERVICE);
35         View view = inflater.inflate(R.layout.preference_seekbar, parent, false);
36     }
37 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\SeekBarPreference.java

```
36 mSeekBar = (SeekBar) view.findViewById(R.id.seekbar);
37 mSeekBar.setProgress(mProgress);
38 mSeekBar.setOnSeekBarChangeListener(this);
39 mSeekBar.setMax(25);
40 return view;
41 }
42
43 @Override
44 public void onProgressChanged(SearchBar seekBar, int progress, boolean fromUser) {
45     if (!fromUser)
46         return;
47     if (progress == 0) {
48         progress = progress + 1;
49     }
50     setValue(progress);
51 }
52
53 @Override
54 public void onStartTrackingTouch(SearchBar seekBar) {
55 }
56
57 @Override
58 public void onStopTrackingTouch(SearchBar seekBar) {
59 }
60
61 @Override
62 protected void onSetInitialValue(boolean restoreValue, Object defaultValue) {
63     setValue(restoreValue ? getPersistedInt(mProgress) : (Integer) defaultValue);
64 }
65
66 public void setValue(int value) {
67     if (shouldPersist()) {
68         persistInt(value);
69     }
70     if (value != mProgress) {
71         --
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\SeekBarPreference.java

```
71     mProgress = value;
72     notifyChanged();
73     }
74     }
75
76 }
77
78
79
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ServiceoClassificacao.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.os.Parcel;
4 import android.os.Parcelable;
5
6 /**
7  * Created by Jucemar on 14/02/2016.
8  */
9 public class ServiceoClassificacao implements Parcelable {
10     private String codigoServico;
11     private String descricaoServico;
12     private String descricaoClassificacao;
13     private String codigoClassificacao;
14
15     public ServiceoClassificacao(String codigoServico, String descricaoServico, String codigoClassificacao, String
descricaoClassificacao) {
16         this.descricaoServico = descricaoServico;
17         this.codigoClassificacao = codigoClassificacao;
18         this.codigoServico = codigoServico;
19         this.descricaoClassificacao = descricaoClassificacao;
20     }
21
22     protected ServiceoClassificacao(Parcel in) {
23         codigoServico = in.readString();
24         descricaoServico = in.readString();
25         descricaoClassificacao = in.readString();
26         codigoClassificacao = in.readString();
27     }
28
29     public static final Creator<ServiceoClassificacao> CREATOR = new Creator<ServiceoClassificacao>() {
30         @Override
31         public ServiceoClassificacao createFromParcel(Parcel in) {
32             return new ServiceoClassificacao(in);
33         }
34     }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ServiceoClassificacao.java

```
35     @Override
36     public ServiceoClassificacao[] newArray(int size) {
37         return new ServiceoClassificacao[size];
38     }
39 };
40
41 public ServiceoClassificacao(String descricaoServico, String descricaoClassificacao) {
42     this.descricaoClassificacao = descricaoClassificacao;
43     this.descricaoServico = descricaoServico;
44 }
45
46 @Override
47 public int describeContents() {
48     return 0;
49 }
50
51 @Override
52 public void writeToParcel(Parcel dest, int flags) {
53     dest.writeString(codigoServico);
54     dest.writeString(descricaoServico);
55     dest.writeString(descricaoClassificacao);
56     dest.writeString(codigoClassificacao);
57 }
58
59 public String getCodigoServico() {
60     return codigoServico;
61 }
62
63 public void setCodigoServico(String codigoServico) {
64     this.codigoServico = codigoServico;
65 }
66
67 public String getDescricaoServico() {
68     return descricaoServico;
69 }
```

```
70
71     public void setDescricaoServico(String descricaoServico) {
72         this.descricaoServico = descricaoServico;
73     }
74
75     public String getDescricaoClassificacao() {
76         return descricaoClassificacao;
77     }
78
79     public void setDescricaoClassificacao(String descricaoClassificacao) {
80         this.descricaoClassificacao = descricaoClassificacao;
81     }
82
83     public String getCodigoClassificacao() {
84         return codigoClassificacao;
85     }
86
87     public void setCodigoClassificacao(String codigoClassificacao) {
88         this.codigoClassificacao = codigoClassificacao;
89     }
90 }
91
92
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ServiceClassificacaoDAO.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.util.Log;
7
8 import java.util.ArrayList;
9 import java.util.Set;
10
11 /**
12  * Created by Jucemar on 19/03/2016.
13  */
14 public class ServiceClassificacaoDAO {
15
16     private static ServiceClassificacaoDAO instancia;
17     private SQLiteDatabase bd;
18
19     public static ServiceClassificacaoDAO getInstance(Context contexto) {
20         if (instancia == null) {
21             instancia = new ServiceClassificacaoDAO(contexto);
22         }
23         return instancia;
24     }
25
26     private ServiceClassificacaoDAO(Context contexto) {
27         BancoDeDados bancoDeDados = BancoDeDados.getInstance(contexto);
28         bd = bancoDeDados.getWritableDatabase();
29     }
30
31     public ArrayList<ServiceClassificacao> listarServicosEspecializados(String cnes) {
32         String[] argumentos = {cnes};
33         String query = "SELECT s.descricao as servico, c.descricao as classificacao FROM servicos JOIN classificacao c
34         ON s.id=c.servico_id JOIN servico_classificacao estabelecimento ON servico_classificacao_estabelecimento.
35         classificacao_id=c.id WHERE servico_classificacao_estabelecimento.estabelecimento_cnes=?";
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ServiceoClassificacaoDAO.java

```
34 Cursor c = bd.rawQuery(query, argumentos);
35 Log.i("Cursor", String.valueOf(c.getCount()));
36 ArrayList<ServicoClassificacao> resultado = new ArrayList<ServicoClassificacao>();
37 if (c.moveToFirst()) {
38     do {
39         ServicoClassificacao sc = extrairDoCursor(c);
40         resultado.add(sc);
41     } while (c.moveToNext());
42 }
43 c.close();
44 Log.e("ResultadoBuscaProf", String.valueOf(resultado.size()));
45 return resultado;
46 }
47
48 private static ServicoClassificacao extrairDoCursor(Cursor c) {
49     String servico = c.getString(0);
50     String classificacao = c.getString(1);
51     return new ServicoClassificacao(servico.toUpperCase(), classificacao.toUpperCase());
52 }
53
54 public CharSequence[] listarTiposServicos(Set<String> setMunicipios) {
55     CharSequence[] resultado = null;
56     if (setMunicipios.size() > 0) {
57         StringBuffer filtroMunicipios = new StringBuffer();
58         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
59         for (int i = 0; i < municipios.length; i++) {
60             filtroMunicipios.append("'" + municipios[i] + "'");
61             if (i < municipios.length - 1) {
62                 filtroMunicipios.append(", ");
63             }
64         }
65         String query = "SELECT DISTINCT servico.descricao FROM estabelecimento " +
66             "JOIN servico_classificacao_estabelecimento ON estabelecimento.cnes="
67             "servico_classificacao_estabelecimento.estabelecimento_cnes " +
68             "JOIN classificacao ON servico_classificacao_estabelecimento.classificacao_id=classificacao.id " +
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar\dimon\cadesaude\ServiceoClassificacaoDAO.java

```
68         "JOIN servico ON classificacao.servico_id=servico.id WHERE estabelecimento.codigo_municipio IN ("
        + filtroMunicipios + ") ORDER BY servico.descricao;";
69     Cursor c = bd.rawQuery(query, null);
70     resultado = new CharSequence[c.getCount()];
71     int i = 0;
72     if (c.moveToFirst()) {
73         do {
74             resultado[i] = c.getString(c.getColumnIndex(TabelaServico.Colunas.DESCRICAO));
75             i++;
76         } while (c.moveToNext());
77     }
78     c.close();
79 }
80     return resultado;
81 }
82
83     public ArrayList<Estabelecimento> buscaStringEmServicos(Set<String> setMunicipios, String sentenca) {
84         StringBuffer filtroMunicipios = new StringBuffer();
85         String[] municipios = setMunicipios.toArray(new String[setMunicipios.size()]);
86         for (int i = 0; i < municipios.length; i++) {
87             filtroMunicipios.append("'" + municipios[i] + "'");
88             if (i < municipios.length - 1) {
89                 filtroMunicipios.append(", ");
90             }
91         }
92         String query = "SELECT * FROM estabelecimento WHERE cnes IN (SELECT DISTINCT estabelecimento.cnes FROM
93             servico_classificacao_estabelecimento \n" +
94             "JOIN estabelecimento ON servico_classificacao_estabelecimento.estabelecimento_cnes=estabelecimento.
95             cnes\n" +
96             "JOIN classificacao ON classificacao.id=servico_classificacao_estabelecimento.classificacao_id\n" +
97             "JOIN servico ON servico.id=classificacao.servico_id\n" +
98             "WHERE (servico.descricao LIKE '%" + sentenca + "%' OR classificacao.descricao LIKE '%" + sentenca +
99             "%')) AND estabelecimento.codigo_municipio IN (" + filtroMunicipios + "));";
100     ArrayList<Estabelecimento> resultado = new ArrayList<Estabelecimento>();
101     Cursor c = bd.rawQuery(query, null);
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ServiceClassificacaoDAO.java

```
99  if (c.moveToFirst()) {
100      do {
101          Estabelecimento e = EstabelecimentoDAO.extrairDoCursor(c);
102          resultado.add(e);
103      } while (c.moveToNext());
104  }
105  c.close();
106  return resultado;
107  }
108
109 }
110
111
112
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ServicesAdapter.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.Context;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.AdapterView;
8 import android.widget.TextView;
9
10 import java.util.ArrayList;
11
12 /**
13  * Created by Jucemar on 19/03/2016.
14  */
15 public class ServicosAdapter extends ArrayAdapter<ServicoClassificacao> {
16
17
18     public ServicosAdapter(Context context, ArrayList<ServicoClassificacao> objects) {
19         super(context, R.layout.fragment_listas, objects);
20     }
21
22     @Override
23     public View getView(int position, View convertView, ViewGroup parent) {
24         ServicoClassificacao sc = getItem(position);
25         if (convertView == null) {
26             convertView = LayoutInflater.from(getContext()).inflate(R.layout.item_listas, parent, false);
27
28             TextView tvNome = (TextView) convertView.findViewById(R.id.item_linha_1);
29             TextView tvCargo = (TextView) convertView.findViewById(R.id.item_linha_2);
30             tvNome.setText(sc.getDescricaoServico());
31             tvCargo.setText(sc.getDescricaoClassificacao());
32             return convertView;
33         }
34     }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\ServicesFragment.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.os.Bundle;
4 import android.support.v4.app.Fragment;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.ListView;
9
10 import java.util.ArrayList;
11
12 /**
13  * Created by Jucemar on 18/03/2016.
14  */
15 public class ServicosFragment extends Fragment {
16
17     @Override
18     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
19         View v = inflater.inflate(R.layout.fragment_listas, container, false);
20         ArrayList<ServicoClassificacao> listaDeServicos = getArguments().getParcelableArrayList("listaDeServicos");
21         ListView lv = (ListView) v.findViewById(R.id.list_view);
22         lv.setDivider(null);
23         if (listaDeServicos.size() > 0) {
24             ServicosAdapter sa = new ServicosAdapter(getActivity(), listaDeServicos);
25             lv.setAdapter(sa);
26         } else {
27             ArrayList<ServicoClassificacao> erro = new ArrayList<ServicoClassificacao>();
28             erro.add(new ServicoClassificacao("Não há informações cadastradas", "Desculpe o transtorno"));
29             ServicosAdapter sa = new ServicosAdapter(getActivity(), erro);
30             lv.setAdapter(sa);
31         }
32         return v;
33     }
34 }
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\SobreActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 import android.content.pm.PackageInfo;
4 import android.content.pm.PackageManager;
5 import android.os.Bundle;
6 import android.support.v7.app.AppCompatActivity;
7 import android.support.v7.widget.Toolbar;
8 import android.view.MenuItem;
9 import android.widget.TextView;
10
11 public class SobreActivity extends AppCompatActivity {
12
13     private Toolbar toolbar;
14     private TextView txvAppVersao;
15     private TextView txvBdVersao;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_sobre);
21         toolbar = (Toolbar) findViewById(R.id.toolbar_actvt_sobre);
22         setSupportActionBar(toolbar);
23         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
24         toolbar.setTitle("Sobre");
25         txvAppVersao = (TextView) findViewById(R.id.sobre_txv_app_versao);
26         txvBdVersao = (TextView) findViewById(R.id.sobre_txv_bd_versao);
27         PackageInfo pInfo = null;
28         try {
29             pInfo = getPackageManager().getPackageInfo(getPackageName(), 0);
30         } catch (PackageManager.NameNotFoundException e) {
31             e.printStackTrace();
32         }
33         String appVersao = pInfo.versionName;
34         txvAppVersao.setText(getString(R.string.appVersao) + " " + appVersao);
35         txvBdVersao.setText(getString(R.string.bdVersao) + " " + BancoDeDados.BD_VERSAO);
36     }
37 }
```


File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\SobreActivity.java

```
36     }
37
38     @Override
39     public boolean onOptionsItemSelected(MenuItem item) {
40         int id = item.getItemId();
41         switch (id) {
42             case android.R.id.home:
43                 onBackPressed();
44                 break;
45             }
46         return super.onOptionsItemSelected(item);
47     }
48 }
49 }
50
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juccemar_dimon\cadesaude\StringUtils.java

```
1 package br.com.juccemar_dimon.cadesaude;
2
3 import android.util.Log;
4
5 import java.text.Normalizer;
6
7 /**
8  * Created by Juccemar on 20/04/2016.
9  */
10 public class StringUtils {
11
12     public static String removerAcentos(String str) {
13         CharSequence cs = new StringBuilder(str == null ? "" : str);
14         return Normalizer.normalize(cs, Normalizer.Form.NFKD).replaceAll("\\p{InCombiningDiacriticalMarks}+", "");
15     }
16
17     public static String formatarTelefones(String telefoneSujo) {
18         StringBuffer telefoneFinal = new StringBuffer();
19         StringBuffer telefoneLimpo = new StringBuffer();
20         for (int i = 0; i < telefoneSujo.length(); i++) {
21             String s = telefoneSujo.charAt(i) + "";
22             try {
23                 Integer.parseInt(s);
24                 telefoneLimpo.append(s);
25             } catch (NumberFormatException e) {
26                 Log.e("Telefone", "Caractere inválido no telefone: " + s);
27             }
28         }
29         Log.e("Telefone", "telefone: " + telefoneLimpo.toString());
30         if (telefoneLimpo.length() == 10) {
31             String fim = (String) telefoneLimpo.subSequence(2, 10);
32             String inicio = (String) telefoneLimpo.subSequence(0, 2);
33             if (telefoneLimpo.charAt(2) == '8' || telefoneLimpo.charAt(2) == '9') {
34                 telefoneFinal.append(inicio);
35                 telefoneFinal.append("9");
36             }
37         }
38     }
39 }
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juceimar_dimon\cadesaude\StringUtils.java

```
36     telefoneFinal.append(fim);
37     Log.e("Telefone", inicio.toString());
38     Log.e("Telefone", fim.toString());
39     Log.e("Telefone", telefoneFinal.toString());
40 } else {
41     telefoneFinal.append("0");
42     telefoneFinal.append(inicio);
43     telefoneFinal.append(fim);
44 }
45 } else {
46     telefoneFinal = telefoneLimpo;
47 }
48 return telefoneFinal.toString();
49 }
50 }
51 }
52 }
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\juccemar_dimon\cadesaude\TabelaAtendimento.java

```
1 package br.com.juccemar_dimon.cadesaude;
2
3 /**
4  * Created by Juccemar on 10/03/2016.
5  */
6 public final class TabelaAtendimento {
7     public static final String NOME_TABELA = "atendimento_estabelecimento";
8
9     public static final class Colunas {
10         public static final String ID = "id";
11         public static final String ATENDIMENTO = "atendimento";
12         public static final String CONVENIO = "convenio";
13         public static final String ESTABELECIMENTO_CNES = "estabelecimento_cnes";
14     }
15
16 }
17
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\TabelaClassificacao.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4  * Created by Jucemar on 10/03/2016.
5  */
6 public final class TabelaClassificacao {
7     public static final String NOME_TABELA = "classificacao";
8
9     public static final class Colunas {
10         public static final String ID = "id";
11         public static final String SERVICO_ID = "servico_id";
12         public static final String CODIGO = "codigo";
13         public static final String DESCRICAO = "descricao";
14     }
15
16 }
17
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\juccemar_dimon\cadesaude\TabelaEstabelecimento.java

```
1 package br.com.juccemar_dimon.cadesaude;
2
3 /**
4  * Created by Juccemar on 10/03/2016.
5  */
6 public final class TabelaEstabelecimento {
7     public static final String NOME_TABELA = "estabelecimento";
8
9     public static final class Colunas {
10         public static final String CNES = "cnes";
11         public static final String RAZAO_SOCIAL = "razao_social";
12         public static final String NOME_FANTASIA = "nome_fantasia";
13         public static final String TELEFONE = "telefone";
14         public static final String NOME_LOGRADOURO = "nome_logradouro";
15         public static final String NUMERO = "numero";
16         public static final String COMPLEMENTO = "complemento";
17         public static final String BAIRRO = "bairro";
18         public static final String MUNICIPIO = "municipio";
19         public static final String CODIGO_MUNICIPIO = "codigo_municipio";
20         public static final String ESTADO = "estado";
21         public static final String CEP = "cep";
22         public static final String LATITUDE = "latitude";
23         public static final String LONGITUDE = "longitude";
24         public static final String TIPO_ESTABELECIMENTO = "tipo_estabelecimento";
25     }
26 }
27
28
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\TabelaProfissionais.java

```
1 package br.com.jucemar_dimon.cadesaude;
2
3 /**
4  * Created by Jucemar on 10/03/2016.
5  */
6 public final class TabelaProfissionais {
7     public static final String NOME_TABELA="profissional";
8
9     public static final class Colunas{
10         public static final String ID="id";
11         public static final String NOME="nome";          CBO="cbo";
12         public static final String CBO_DESCRICA0="cbo_descricao";
13         public static final String
14
15     }
16
17
18
19
20 }
21
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\juccemar_dimon\cadesaude\TabelaServico.java

```
1 package br.com.juccemar_dimon.cadesaude;
2
3 /**
4  * Created by Juccemar on 10/03/2016.
5  */
6 public final class TabelaServico {
7     public static final String NOME_TABELA = "servico";
8
9     public static final class Colunas {
10         public static final String ID = "id";
11         public static final String CODIGO = "codigo";
12         public static final String DESCRICAO = "descricao";
13     }
14 }
15 }
16
```


File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\juccemar_dimon\cadesaude\TabelaServicoClassificacao.java

```
1 package br.com.juccemar_dimon.cadesaude;
2
3 /**
4  * Created by Juccemar on 10/03/2016.
5  */
6 public final class TabelaServicoClassificacao {
7     public static final String NOME_TABELA = "servico_classificacao_estabelecimento";
8
9     public static final class Colunas {
10         public static final String ESTABELECIMENTO_CNES = "estabelecimento_cnes";
11         public static final String CLASSIFICACAO_ID = "classificacao_id";
12     }
13
14 }
15
```

File - E:\Android\Projetos\CadESaude\app\src\main\java\br\com\juccemar_dimon\cadesaude\TabelaVinculosProfissional.java

```
1 package br.com.juccemar_dimon.cadesaude;
2
3 /**
4  * Created by Juccemar on 10/03/2016.
5  */
6 public final class TabelaVinculosProfissional {
7     public static final String NOME_TABELA = "vinculo_profissional_estabelecimento";
8
9     public static final class Colunas {
10         public static final String ID = "id";
11         public static final String ESTABELECI_MENTO_CNES = "estabelecimento_cnes";
12     }
13
14 }
15
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\TelaInicialActivity.java

```
1 package br.com.jucemar_dimon.cadesaude;
2 /**
3  * Created by Jucemar on 10/03/2016.
4  */
5
6 import android.content.Intent;
7 import android.content.pm.PackageInfo;
8 import android.content.pm.PackageManager;
9 import android.os.Bundle;
10 import android.os.Handler;
11 import android.preference.PreferenceManager;
12 import android.support.v7.app.AppCompatActivity;
13 import android.widget.TextView;
14
15 import java.util.HashSet;
16 import java.util.Set;
17
18 public class TelaInicialActivity extends AppCompatActivity {
19
20     private static int SPLASH_TIME_OUT = 2000;
21     private Set<String> listaDeMunicipios;
22     private TextView txtv_versao;
23
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_tela_inicial);
28         txtv_versao = (TextView) findViewById(R.id.txtv_versao_app);
29         PackageInfo pInfo = null;
30         try {
31             pInfo = getPackageManager().getPackageInfo(getPackageName(), 0);
32         } catch (PackageManager.NameNotFoundException e) {
33             e.printStackTrace();
34         }
35         String appVersao = pInfo.versionName;
```

File - E:\Android\Projetos\CadEsaude\app\src\main\java\br\com\jucemar_dimon\cadesaude\TelainicialActivity.java

```
36     txv_versao.setText("Versão " + appVersao);
37     new Handler().postDelayed(new Runnable() {
38         /*
39          * Exibindo a tela inicial.
40          */
41         @Override
42         public void run() {
43             int qtde = PreferenceManager.getDefaultSharedPreferences(getApplicationContext())
44                 .getString(R.string.pref_municipios_preferidos key), new HashSet<String>()).size()
45             ;
46             // Esse método será executado sempre que o tempo acabar
47             // E inicia a activity principal
48             if (qtde > 0) {
49                 Intent i = new Intent(TelainicialActivity.this, MainActivity.class);
50                 startActivity(i);
51             } else {
52                 Intent i = new Intent(TelainicialActivity.this, PrimeirosPassosActivity.class);
53                 startActivity(i);
54             }
55             // Fecha a tela de apresentação
56             finish();
57             }, SPLASH_TIME_OUT);
58         }
59     @Override
60     public void onBackPressed() {
61         System.exit(0);
62     }
63 }
64 }
65 //
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda.xml

```
1 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:app="http://schemas.android.com/apk/res-auto"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent">
5
6   <android.support.design.widget.AppBarLayout
7       android:layout_width="match_parent"
8       android:layout_height="wrap_content"
9       android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
10
11       <android.support.v7.widget.Toolbar
12           android:id="@+id/toolbar_actvt_ajuda"
13           android:layout_width="match_parent"
14           android:layout_height="?attr/actionBarSize"
15           android:background="?attr/colorPrimary"
16           app:layout_scrollFlags="scroll|enterAlways"
17           app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
18   </android.support.design.widget.AppBarLayout>
19
20   <android.support.v4.widget.NestedScrollView
21       android:id="@+id/nestedScroll"
22       android:layout_width="match_parent"
23       android:layout_height="match_parent"
24       android:scrollbars="vertical"
25       app:layout_behavior="@string/appbar_scrolling_view_behavior">
26
27       <LinearLayout
28           android:layout_width="match_parent"
29           android:layout_height="match_parent"
30           android:layout_margin="16dp"
31           android:orientation="vertical">
32
33           <ImageView
34               android:layout_width="wrap_content"
35               android:layout_height="wrap_content"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda.xml

```
36         android:layout_gravity="center"
37         android:layout_margin="16dp"
38         android:src="@drawable/ic_help_circle_outline_grey600_48dp" />
39
40     <TextView
41         android:layout_width="wrap_content"
42         android:layout_height="wrap_content"
43         android:layout_margin="16dp"
44         android:text="@string/cabecalho_ajuda" />
45
46     <Button
47         android:id="@+id/btn_tipos_estabelecimentos"
48         android:layout_width="match_parent"
49         android:layout_height="wrap_content"
50         android:layout_margin="16dp"
51         android:text="@string/label_btn_tipos_estabelecimentos" />
52
53     <Button
54         android:id="@+id/btn_tipos_servicos"
55         android:layout_width="match_parent"
56         android:layout_height="wrap_content"
57         android:layout_margin="16dp"
58         android:text="@string/label_btn_tipos_servicos" />
59     </LinearLayout>
60     </android.support.v4.widget.NestedScrollView>
61 </android.support.design.widget.CoordinatorLayout>
62 </>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
1 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:app="http://schemas.android.com/apk/res-auto"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent">
5
6   <android.support.design.widget.AppBarLayout
7     android:layout_width="match_parent"
8     android:layout_height="wrap_content"
9     android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
10
11     <android.support.v7.widget.Toolbar
12       android:id="@+id/toolbar_actvt_ajuda_tipos_servicos"
13       android:layout_width="match_parent"
14       android:layout_height="?attr/actionBarSize"
15       android:background="?attr/colorPrimary"
16       app:layout_scrollFlags="scroll|enterAlways"
17       app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
18
19   </android.support.design.widget.AppBarLayout>
20
21   <android.support.v4.widget.NestedScrollView
22     android:id="@+id/nestedScroll"
23     android:layout_width="match_parent"
24     android:layout_height="match_parent"
25     android:scrollbars="vertical"
26     app:layout_behavior="@string/appbar_scrolling_view_behavior">
27
28     <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
29       android:layout_width="match_parent"
30       android:layout_height="match_parent"
31       android:layout_margin="16dp"
32       android:paddingBottom="32dp"
33       android:shrinkColumns="1">
34
35       <TableRow
```

```
36     android:layout_width="match_parent"
37     android:layout_height="wrap_content"
38     android:background="@color/primary_light"
39     android:gravity="center">
40
41     <TextView
42         android:layout_margin="8dp"
43         android:layout_weight="2"
44         android:width="0dp"
45         android:singleLine="false"
46         android:text="Serviço especializado"
47         android:textStyle="bold" />
48
49     <TextView
50         android:layout_margin="8dp"
51         android:layout_weight="3"
52         android:width="0dp"
53         android:text="Detalhes do serviço"
54         android:textStyle="bold" />
55     </TableRow>
56
57     <TableRow
58         android:layout_width="match_parent"
59         android:layout_height="wrap_content"
60         android:background="#e0e0e0"
61         android:gravity="center">
62
63     <TextView
64         android:layout_margin="8dp"
65         android:layout_weight="2"
66         android:width="0dp"
67         android:singleLine="false"
68         android:text="@string/locl" />
69     <TextView
```


File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
71         android:layout_margin="8dp"
72         android:layout_weight="3"
73         android:width="0dp"
74         android:text="@string/101c2" />
75     </TableRow>
76
77     <TableRow
78         android:layout_width="match_parent"
79         android:layout_height="wrap_content"
80         android:background="@color/primary_light"
81         android:gravity="center">
82
83         <TextView
84             android:layout_margin="8dp"
85             android:layout_weight="2"
86             android:width="0dp"
87             android:text="@string/102c1" />
88
89         <TextView
90             android:layout_margin="8dp"
91             android:layout_weight="3"
92             android:width="0dp"
93             android:text="@string/102c2" />
94     </TableRow>
95
96     <TableRow
97         android:layout_width="match_parent"
98         android:layout_height="wrap_content"
99         android:background="#e0e0e0"
100        android:gravity="center">
101
102        <TextView
103            android:layout_margin="8dp"
104            android:layout_weight="2"
105            android:width="0dp"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
106         android:text="@string/103c1" />
107
108     <TextView
109         android:layout_margin="8dp"
110         android:layout_weight="3"
111         android:width="0dp"
112         android:text="@string/103c2" />
113     </TableRow>
114
115     <TableRow
116         android:layout_width="match_parent"
117         android:layout_height="wrap_content"
118         android:background="@color/primary_light"
119         android:gravity="center">
120
121     <TextView
122         android:layout_margin="8dp"
123         android:layout_weight="2"
124         android:width="0dp"
125         android:text="@string/104c1" />
126
127     <TextView
128         android:layout_margin="8dp"
129         android:layout_weight="3"
130         android:width="0dp"
131         android:text="@string/104c2" />
132     </TableRow>
133
134     <TableRow
135         android:layout_width="match_parent"
136         android:layout_height="wrap_content"
137         android:background="#e0e0e0"
138         android:gravity="center">
139     <TextView
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
141         android:layout_margin="8dp"
142         android:layout_weight="2"
143         android:width="0dp"
144         android:text="@string/105c1" />
145
146     <TextView
147         android:layout_margin="8dp"
148         android:layout_weight="3"
149         android:width="0dp"
150         android:text="@string/105c2" />
151
152 </TableRow>
153
154 <TableRow
155     android:layout_width="match_parent"
156     android:layout_height="wrap_content"
157     android:background="@color/primary_light"
158     android:gravity="center">
159
160     <TextView
161         android:layout_margin="8dp"
162         android:layout_weight="2"
163         android:width="0dp"
164         android:text="@string/106c1" />
165
166     <TextView
167         android:layout_margin="8dp"
168         android:layout_weight="3"
169         android:width="0dp"
170         android:text="@string/106c2" />
171
172 </TableRow>
173
174 <TableRow
175     android:layout_width="match_parent"
176     android:layout_height="wrap_content"
177     android:background="#e0e0e0">
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
176         android:gravity="center">
177
178     <TextView
179         android:layout_margin="8dp"
180         android:layout_weight="2"
181         android:width="0dp"
182         android:text="@string/107c1" />
183
184     <TextView
185         android:layout_margin="8dp"
186         android:layout_weight="3"
187         android:width="0dp"
188         android:text="@string/107c2" />
189
190     <TableRow>
191
192         android:layout_width="match_parent"
193         android:layout_height="wrap_content"
194         android:background="@color/primary_light"
195         android:gravity="center">
196
197     <TextView
198         android:layout_margin="8dp"
199         android:layout_weight="2"
200         android:width="0dp"
201         android:text="@string/108c1" />
202
203     <TextView
204         android:layout_margin="8dp"
205         android:layout_weight="3"
206         android:width="0dp"
207         android:text="@string/108c2" />
208
209     <TableRow>
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
211         android:layout_width="match_parent"
212         android:layout_height="wrap_content"
213         android:background="#e0e0e0"
214         android:gravity="center">
215
216         <TextView
217             android:layout_margin="8dp"
218             android:layout_weight="2"
219             android:width="0dp"
220             android:text="@string/109c1" />
221
222         <TextView
223             android:layout_margin="8dp"
224             android:layout_weight="3"
225             android:width="0dp"
226             android:text="@string/109c2" />
227     </TableRow>
228
229     <TableRow
230         android:layout_width="match_parent"
231         android:layout_height="wrap_content"
232         android:background="@color/primary_light"
233         android:gravity="center">
234
235         <TextView
236             android:layout_margin="8dp"
237             android:layout_weight="2"
238             android:width="0dp"
239             android:text="@string/110c1" />
240
241         <TextView
242             android:layout_margin="8dp"
243             android:layout_weight="3"
244             android:width="0dp"
245             android:text="@string/110c2" />
```

```
246 </TableRow>
247
248 <TableRow
249     android:layout_width="match_parent"
250     android:layout_height="wrap_content"
251     android:background="#e0e0e0"
252     android:gravity="center">
253
254     <TextView
255         android:layout_margin="8dp"
256         android:layout_weight="2"
257         android:width="0dp"
258         android:text="@string/l11c1" />
259
260     <TextView
261         android:layout_margin="8dp"
262         android:layout_weight="3"
263         android:width="0dp"
264         android:text="@string/l11c2" />
265 </TableRow>
266
267 <TableRow
268     android:layout_width="match_parent"
269     android:layout_height="wrap_content"
270     android:background="@color/primary_light"
271     android:gravity="center">
272
273     <TextView
274         android:layout_margin="8dp"
275         android:layout_weight="2"
276         android:width="0dp"
277         android:text="@string/l12c1" />
278
279     <TextView
280         android:layout_margin="8dp"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
281         android:layout_weight="3"
282         android:width="0dp"
283         android:text="@string/112c2" />
284     </TableRow>
285
286     <TableRow
287         android:layout_width="match_parent"
288         android:layout_height="wrap_content"
289         android:background="#e0e0e0"
290         android:gravity="center">
291
292         <TextView
293             android:layout_margin="8dp"
294             android:layout_weight="2"
295             android:width="0dp"
296             android:text="@string/113c1" />
297
298         <TextView
299             android:layout_margin="8dp"
300             android:layout_weight="3"
301             android:width="0dp"
302             android:text="@string/113c2" />
303     </TableRow>
304
305     <TableRow
306         android:layout_width="match_parent"
307         android:layout_height="wrap_content"
308         android:background="@color/primary_light"
309         android:gravity="center">
310
311         <TextView
312             android:layout_margin="8dp"
313             android:layout_weight="2"
314             android:width="0dp"
315             android:text="@string/114c1" />
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
316
317
318         android:layout_margin="8dp"
319         android:layout_weight="3"
320         android:width="0dp"
321         android:text="@string/l14c2" />
322     </TableRow>
323
324     <TableRow
325         android:layout_width="match_parent"
326         android:layout_height="wrap_content"
327         android:background="#e0e0e0"
328         android:gravity="center">
329
330         <TextView
331             android:layout_margin="8dp"
332             android:layout_weight="2"
333             android:width="0dp"
334             android:text="@string/l15c1" />
335
336         <TextView
337             android:layout_margin="8dp"
338             android:layout_weight="3"
339             android:width="0dp"
340             android:text="@string/l15c2" />
341     </TableRow>
342
343     <TableRow
344         android:layout_width="match_parent"
345         android:layout_height="wrap_content"
346         android:background="@color/primary_light"
347         android:gravity="center">
348
349         <TextView
350             android:layout_margin="8dp"
```


File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
351         android:layout_weight="2"
352         android:width="0dp"
353         android:text="@string/116c1" />
354
355     <TextView
356         android:layout_margin="8dp"
357         android:layout_weight="3"
358         android:width="0dp"
359         android:text="@string/116c2" />
360 </TableRow>
361
362 <TableRow
363     android:layout_width="match_parent"
364     android:layout_height="wrap_content"
365     android:background="#e0e0e0"
366     android:gravity="center">
367
368     <TextView
369         android:layout_margin="8dp"
370         android:layout_weight="2"
371         android:width="0dp"
372         android:text="@string/117c1" />
373
374     <TextView
375         android:layout_margin="8dp"
376         android:layout_weight="3"
377         android:width="0dp"
378         android:text="@string/117c2" />
379 </TableRow>
380
381 <TableRow
382     android:layout_width="match_parent"
383     android:layout_height="wrap_content"
384     android:background="@color/primary_light"
385     android:gravity="center">
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
386
387
388         android:layout_margin="8dp"
389         android:layout_weight="2"
390         android:width="0dp"
391         android:text="@string/l18c1" />
392
393     <TextView
394         android:layout_margin="8dp"
395         android:layout_weight="3"
396         android:width="0dp"
397         android:text="@string/l18c2" />
398     </TableRow>
399
400     <TableRow
401         android:layout_width="match_parent"
402         android:layout_height="wrap_content"
403         android:background="#e0e0e0"
404         android:gravity="center">
405
406         <TextView
407             android:layout_margin="8dp"
408             android:layout_weight="2"
409             android:width="0dp"
410             android:text="@string/l19c1" />
411
412         <TextView
413             android:layout_margin="8dp"
414             android:layout_weight="3"
415             android:width="0dp"
416             android:text="@string/l19c2" />
417     </TableRow>
418
419     <TableRow
420         android:layout_width="match_parent"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
421         android:layout_height="wrap_content"
422         android:background="@color/primary_light"
423         android:gravity="center">
424
425     <TextView
426         android:layout_margin="8dp"
427         android:layout_weight="2"
428         android:width="0dp"
429         android:text="@string/l20c1" />
430
431     <TextView
432         android:layout_margin="8dp"
433         android:layout_weight="3"
434         android:width="0dp"
435         android:text="@string/l20c2" />
436
437 </TableRow>
438
439 <TableRow
440     android:layout_width="match_parent"
441     android:layout_height="wrap_content"
442     android:background="#e0e0e0"
443     android:gravity="center">
444
445     <TextView
446         android:layout_margin="8dp"
447         android:layout_weight="2"
448         android:width="0dp"
449         android:text="@string/l21c1" />
450
451     <TextView
452         android:layout_margin="8dp"
453         android:layout_weight="3"
454         android:width="0dp"
455         android:text="@string/l21c2" />
456
457 </TableRow>
```

```
456
457
458     android:layout_width="match_parent"
459     android:layout_height="wrap_content"
460     android:background="@color/primary_light"
461     android:gravity="center">
462
463     <TextView
464         android:layout_margin="8dp"
465         android:layout_weight="2"
466         android:width="0dp"
467         android:text="@string/l22c1" />
468
469     <TextView
470         android:layout_margin="8dp"
471         android:layout_weight="3"
472         android:width="0dp"
473         android:text="@string/l22c2" />
474 </TableRow>
475
476 <TableRow
477     android:layout_width="match_parent"
478     android:layout_height="wrap_content"
479     android:background="#e0e0e0"
480     android:gravity="center">
481
482     <TextView
483         android:layout_margin="8dp"
484         android:layout_weight="2"
485         android:width="0dp"
486         android:text="@string/l23c1" />
487
488     <TextView
489         android:layout_margin="8dp"
490         android:layout_weight="3"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
491         android:width="0dp"
492         android:text="@string/123c2" />
493     </TableRow>
494
495     <TableRow
496         android:layout_width="match_parent"
497         android:layout_height="wrap_content"
498         android:background="@color/primary_light"
499         android:gravity="center">
500
501         <TextView
502             android:layout_margin="8dp"
503             android:layout_weight="2"
504             android:width="0dp"
505             android:text="@string/124c1" />
506
507         <TextView
508             android:layout_margin="8dp"
509             android:layout_weight="3"
510             android:width="0dp"
511             android:text="@string/124c2" />
512     </TableRow>
513
514     <TableRow
515         android:layout_width="match_parent"
516         android:layout_height="wrap_content"
517         android:background="#e0e0e0"
518         android:gravity="center">
519
520         <TextView
521             android:layout_margin="8dp"
522             android:layout_weight="2"
523             android:width="0dp"
524             android:text="@string/125c1" />
525     </TableRow>
```

```
526 <TextView
527     android:layout_margin="8dp"
528     android:layout_weight="3"
529     android:width="0dp"
530     android:text="@string/125c2" />
531 </TableRow>
532
533 <TableRow
534     android:layout_width="match_parent"
535     android:layout_height="wrap_content"
536     android:background="@color/primary_light"
537     android:gravity="center">
538
539 <TextView
540     android:layout_margin="8dp"
541     android:layout_weight="2"
542     android:width="0dp"
543     android:text="@string/126c1" />
544
545 <TextView
546     android:layout_margin="8dp"
547     android:layout_weight="3"
548     android:width="0dp"
549     android:text="@string/126c2" />
550 </TableRow>
551
552 <TableRow
553     android:layout_width="match_parent"
554     android:layout_height="wrap_content"
555     android:background="#e0e0e0"
556     android:gravity="center">
557
558 <TextView
559     android:layout_margin="8dp"
560     android:layout_weight="2"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
561         android:width="0dp"
562         android:singleLine="false"
563         android:text="@string/127c1" />
564
565     <TextView
566         android:layout_margin="8dp"
567         android:layout_weight="3"
568         android:width="0dp"
569         android:singleLine="false"
570         android:text="@string/127c2" />
571
572 </TableRow>
573
574 <TableRow
575     android:layout_width="match_parent"
576     android:layout_height="wrap_content"
577     android:background="@color/primary_light"
578     android:gravity="center">
579
580     <TextView
581         android:layout_margin="8dp"
582         android:layout_weight="2"
583         android:width="0dp"
584         android:text="@string/128c1" />
585
586     <TextView
587         android:layout_margin="8dp"
588         android:layout_weight="3"
589         android:width="0dp"
590         android:text="@string/128c2" />
591
592 </TableRow>
593
594 <TableRow
595     android:layout_width="match_parent"
596     android:layout_height="wrap_content"
597     android:background="#e0e0e0"
598     --
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
596         android:gravity="center">
597
598     <TextView
599         android:layout_margin="8dp"
600         android:layout_weight="2"
601         android:width="0dp"
602         android:text="@string/129c1" />
603
604     <TextView
605         android:layout_margin="8dp"
606         android:layout_weight="3"
607         android:width="0dp"
608         android:text="@string/129c2" />
609
610     <TableRow>
611
612         android:layout_width="match_parent"
613         android:layout_height="wrap_content"
614         android:background="@color/primary_light"
615         android:gravity="center">
616
617     <TextView
618         android:layout_margin="8dp"
619         android:layout_weight="2"
620         android:width="0dp"
621         android:text="@string/130c1" />
622
623     <TextView
624         android:layout_margin="8dp"
625         android:layout_weight="3"
626         android:width="0dp"
627         android:text="@string/130c2" />
628
629     <TableRow>
```


File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
631         android:layout_width="match_parent"
632         android:layout_height="wrap_content"
633         android:background="#e0e0e0"
634         android:gravity="center">
635
636         <TextView
637             android:layout_margin="8dp"
638             android:layout_weight="2"
639             android:width="0dp"
640             android:text="@string/l31c1" />
641
642         <TextView
643             android:layout_margin="8dp"
644             android:layout_weight="3"
645             android:width="0dp"
646             android:text="@string/l31c2" />
647     </TableRow>
648
649     <TableRow
650         android:layout_width="match_parent"
651         android:layout_height="wrap_content"
652         android:background="@color/primary_light"
653         android:gravity="center">
654
655         <TextView
656             android:layout_margin="8dp"
657             android:layout_weight="2"
658             android:width="0dp"
659             android:text="@string/l32c1" />
660
661         <TextView
662             android:layout_margin="8dp"
663             android:layout_weight="3"
664             android:width="0dp"
665             android:text="@string/l32c2" />
666     </TableRow>
```

```
666 </TableRow>
667
668 <TableRow
669     android:layout_width="match_parent"
670     android:layout_height="wrap_content"
671     android:background="#e0e0e0"
672     android:gravity="center">
673
674     <TextView
675         android:layout_margin="8dp"
676         android:layout_weight="2"
677         android:width="0dp"
678         android:text="@string/l33c1" />
679
680     <TextView
681         android:layout_margin="8dp"
682         android:layout_weight="3"
683         android:width="0dp"
684         android:text="@string/l33c2" />
685 </TableRow>
686
687 <TableRow
688     android:layout_width="match_parent"
689     android:layout_height="wrap_content"
690     android:background="@color/primary_light"
691     android:gravity="center">
692
693     <TextView
694         android:layout_margin="8dp"
695         android:layout_weight="2"
696         android:width="0dp"
697         android:text="@string/l34c1" />
698
699     <TextView
700         android:layout_margin="8dp"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
701         android:layout_weight="3"
702         android:width="0dp"
703         android:text="@string/134c2" />
704     </TableRow>
705
706     <TableRow
707         android:layout_width="match_parent"
708         android:layout_height="wrap_content"
709         android:background="#e0e0e0"
710         android:gravity="center">
711
712         <TextView
713             android:layout_margin="8dp"
714             android:layout_weight="2"
715             android:width="0dp"
716             android:text="@string/135c1" />
717
718         <TextView
719             android:layout_margin="8dp"
720             android:layout_weight="3"
721             android:width="0dp"
722             android:text="@string/135c2" />
723     </TableRow>
724
725     <TableRow
726         android:layout_width="match_parent"
727         android:layout_height="wrap_content"
728         android:background="@color/primary_light"
729         android:gravity="center">
730
731         <TextView
732             android:layout_margin="8dp"
733             android:layout_weight="2"
734             android:width="0dp"
735             android:text="@string/136c1" />
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
736
737
738         <TextView
739             android:layout_margin="8dp"
740             android:layout_weight="3"
741             android:width="0dp"
742             android:text="@string/l36c2" />
743     </TableRow>
744
745     <TableRow
746         android:layout_width="match_parent"
747         android:layout_height="wrap_content"
748         android:background="#e0e0e0"
749         android:gravity="center">
750
751         <TextView
752             android:layout_margin="8dp"
753             android:layout_weight="2"
754             android:width="0dp"
755             android:text="@string/l37c1" />
756
757         <TextView
758             android:layout_margin="8dp"
759             android:layout_weight="3"
760             android:width="0dp"
761             android:text="@string/l37c2" />
762     </TableRow>
763
764     <TableRow
765         android:layout_width="match_parent"
766         android:layout_height="wrap_content"
767         android:background="@color/primary_light"
768         android:gravity="center">
769
770         <TextView
771             android:layout_margin="8dp"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
771         android:layout_weight="2"
772         android:width="0dp"
773         android:text="@string/138c1" />
774
775     <TextView
776         android:layout_margin="8dp"
777         android:layout_weight="3"
778         android:width="0dp"
779         android:text="@string/138c2" />
780 </TableRow>
781
782 <TableRow
783     android:layout_width="match_parent"
784     android:layout_height="wrap_content"
785     android:background="#e0e0e0"
786     android:gravity="center">
787
788     <TextView
789         android:layout_margin="8dp"
790         android:layout_weight="2"
791         android:width="0dp"
792         android:text="@string/139c1" />
793
794     <TextView
795         android:layout_margin="8dp"
796         android:layout_weight="3"
797         android:width="0dp"
798         android:text="@string/139c2" />
799 </TableRow>
800
801 <TableRow
802     android:layout_width="match_parent"
803     android:layout_height="wrap_content"
804     android:background="@color/primary_light"
805     android:gravity="center">
```

```
806
807
808         <TextView
809             android:layout_margin="8dp"
810             android:layout_weight="2"
811             android:width="0dp"
812             android:text="@string/140c1" />
813
814         <TextView
815             android:layout_margin="8dp"
816             android:layout_weight="3"
817             android:width="0dp"
818             android:text="@string/140c2" />
819     </TableRow>
820
821     <TableRow
822         android:layout_width="match_parent"
823         android:layout_height="wrap_content"
824         android:background="#e0e0e0"
825         android:gravity="center">
826
827         <TextView
828             android:layout_margin="8dp"
829             android:layout_weight="2"
830             android:width="0dp"
831             android:text="@string/141c1" />
832
833         <TextView
834             android:layout_margin="8dp"
835             android:layout_weight="3"
836             android:width="0dp"
837             android:text="@string/141c2" />
838     </TableRow>
839
840     <TableRow
841         android:layout_width="match_parent"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
841         android:layout_height="wrap_content"
842         android:background="@color/primary_light"
843         android:gravity="center">
844
845     <TextView
846         android:layout_margin="8dp"
847         android:layout_weight="2"
848         android:width="0dp"
849         android:text="@string/l42c1" />
850
851     <TextView
852         android:layout_margin="8dp"
853         android:layout_weight="3"
854         android:width="0dp"
855         android:text="@string/l42c2" />
856 </TableRow>
857
858 <TableRow
859     android:layout_width="match_parent"
860     android:layout_height="wrap_content"
861     android:background="#e0e0e0"
862     android:gravity="center">
863
864     <TextView
865         android:layout_margin="8dp"
866         android:layout_weight="2"
867         android:width="0dp"
868         android:text="@string/l43c1" />
869
870     <TextView
871         android:layout_margin="8dp"
872         android:layout_weight="3"
873         android:width="0dp"
874         android:text="@string/l43c2" />
875 </TableRow>
```

```
876
877
878     android:layout_width="match_parent"
879     android:layout_height="wrap_content"
880     android:background="@color/primary_light"
881     android:gravity="center">
882
883     <TextView
884         android:layout_margin="8dp"
885         android:layout_weight="2"
886         android:width="0dp"
887         android:text="@string/l44c1" />
888
889     <TextView
890         android:layout_margin="8dp"
891         android:layout_weight="3"
892         android:width="0dp"
893         android:text="@string/l44c2" />
894 </TableRow>
895
896 <TableRow
897     android:layout_width="match_parent"
898     android:layout_height="wrap_content"
899     android:background="#e0e0e0"
900     android:gravity="center">
901
902     <TextView
903         android:layout_margin="8dp"
904         android:layout_weight="2"
905         android:width="0dp"
906         android:text="@string/l45c1" />
907
908     <TextView
909         android:layout_margin="8dp"
910         android:layout_weight="3"
```


File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
911         android:width="0dp"
912         android:text="@string/145c2" />
913     </TableRow>
914
915     <TableRow
916         android:layout_width="match_parent"
917         android:layout_height="wrap_content"
918         android:background="@color/primary_light"
919         android:gravity="center">
920
921         <TextView
922             android:layout_margin="8dp"
923             android:layout_weight="2"
924             android:width="0dp"
925             android:text="@string/146c1" />
926
927         <TextView
928             android:layout_margin="8dp"
929             android:layout_weight="3"
930             android:width="0dp"
931             android:text="@string/146c2" />
932     </TableRow>
933
934     <TableRow
935         android:layout_width="match_parent"
936         android:layout_height="wrap_content"
937         android:background="#e0e0e0"
938         android:gravity="center">
939
940         <TextView
941             android:layout_margin="8dp"
942             android:layout_weight="2"
943             android:width="0dp"
944             android:text="@string/147c1" />
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
946 <TextView
947     android:layout_margin="8dp"
948     android:layout_weight="3"
949     android:width="0dp"
950     android:text="@string/147c2" />
951 </TableRow>
952
953 <TableRow
954     android:layout_width="match_parent"
955     android:layout_height="wrap_content"
956     android:background="@color/primary_light"
957     android:gravity="center">
958
959 <TextView
960     android:layout_margin="8dp"
961     android:layout_weight="2"
962     android:width="0dp"
963     android:text="@string/148c1" />
964
965 <TextView
966     android:layout_margin="8dp"
967     android:layout_weight="3"
968     android:width="0dp"
969     android:text="@string/148c2" />
970 </TableRow>
971
972 <TableRow
973     android:layout_width="match_parent"
974     android:layout_height="wrap_content"
975     android:background="#e0e0e0"
976     android:gravity="center">
977
978 <TextView
979     android:layout_margin="8dp"
980     android:layout_weight="2"
```

```
981         android:width="0dp"
982         android:text="@string/149c1" />
983
984     <TextView
985         android:layout_margin="8dp"
986         android:layout_weight="3"
987         android:width="0dp"
988         android:text="@string/149c2" />
989     </TableRow>
990
991     <TableRow
992         android:layout_width="match_parent"
993         android:layout_height="wrap_content"
994         android:background="@color/primary_light"
995         android:gravity="center">
996
997     <TextView
998         android:layout_margin="8dp"
999         android:layout_weight="2"
1000         android:width="0dp"
1001         android:text="@string/150c1" />
1002
1003     <TextView
1004         android:layout_margin="8dp"
1005         android:layout_weight="3"
1006         android:width="0dp"
1007         android:text="@string/150c2" />
1008     </TableRow>
1009
1010     <TableRow
1011         android:layout_width="match_parent"
1012         android:layout_height="wrap_content"
1013         android:background="#e0e0e0"
1014         android:gravity="center">
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
1016 <TextView
1017     android:layout_margin="8dp"
1018     android:layout_weight="2"
1019     android:width="0dp"
1020     android:text="@string/151c1" />
1021
1022 <TextView
1023     android:layout_margin="8dp"
1024     android:layout_weight="3"
1025     android:width="0dp"
1026     android:text="@string/151c2" />
1027 </TableRow>
1028
1029 <TableRow
1030     android:layout_width="match_parent"
1031     android:layout_height="wrap_content"
1032     android:background="@color/primary_light"
1033     android:gravity="center">
1034
1035 <TextView
1036     android:layout_margin="8dp"
1037     android:layout_weight="2"
1038     android:width="0dp"
1039     android:text="@string/152c1" />
1040
1041 <TextView
1042     android:layout_margin="8dp"
1043     android:layout_weight="3"
1044     android:width="0dp"
1045     android:text="@string/152c2" />
1046 </TableRow>
1047
1048 <TableRow
1049     android:layout_width="match_parent"
1050     android:layout_height="wrap_content"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
1051     android:background="#e0e0e0"
1052     android:gravity="center">
1053
1054     <TextView
1055         android:layout_margin="8dp"
1056         android:layout_weight="2"
1057         android:width="0dp"
1058         android:text="@string/153c1" />
1059
1060     <TextView
1061         android:layout_margin="8dp"
1062         android:layout_weight="3"
1063         android:width="0dp"
1064         android:text="@string/153c2" />
1065     </TableRow>
1066
1067     <TableRow
1068         android:layout_width="match_parent"
1069         android:layout_height="wrap_content"
1070         android:background="@color/primary_light"
1071         android:gravity="center">
1072
1073         <TextView
1074             android:layout_margin="8dp"
1075             android:layout_weight="2"
1076             android:width="0dp"
1077             android:text="@string/154c1" />
1078
1079         <TextView
1080             android:layout_margin="8dp"
1081             android:layout_weight="3"
1082             android:width="0dp"
1083             android:text="@string/154c2" />
1084     </TableRow>
1085     -----
```

```
1086 <TableRow
1087     android:layout_width="match_parent"
1088     android:layout_height="wrap_content"
1089     android:background="#e0e0e0"
1090     android:gravity="center">
1091
1092     <TextView
1093         android:layout_margin="8dp"
1094         android:layout_weight="2"
1095         android:width="0dp"
1096         android:text="@string/155c1" />
1097
1098     <TextView
1099         android:layout_margin="8dp"
1100         android:layout_weight="3"
1101         android:width="0dp"
1102         android:text="@string/155c2" />
1103 </TableRow>
1104
1105 <TableRow
1106     android:layout_width="match_parent"
1107     android:layout_height="wrap_content"
1108     android:background="@color/primary_light"
1109     android:gravity="center">
1110
1111     <TextView
1112         android:layout_margin="8dp"
1113         android:layout_weight="2"
1114         android:width="0dp"
1115         android:text="@string/156c1" />
1116
1117     <TextView
1118         android:layout_margin="8dp"
1119         android:layout_weight="3"
1120         android:width="0dp"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
1121         android:text="@string/156c2" />
1122     </TableRow>
1123
1124     <TableRow
1125         android:layout_width="match_parent"
1126         android:layout_height="wrap_content"
1127         android:background="#e0e0e0"
1128         android:gravity="center">
1129
1130         <TextView
1131             android:layout_margin="8dp"
1132             android:layout_weight="2"
1133             android:width="0dp"
1134             android:text="@string/157c1" />
1135
1136         <TextView
1137             android:layout_margin="8dp"
1138             android:layout_weight="3"
1139             android:width="0dp"
1140             android:text="@string/157c2" />
1141     </TableRow>
1142
1143     <TableRow
1144         android:layout_width="match_parent"
1145         android:layout_height="wrap_content"
1146         android:background="@color/primary_light"
1147         android:gravity="center">
1148
1149         <TextView
1150             android:layout_margin="8dp"
1151             android:layout_weight="2"
1152             android:width="0dp"
1153             android:text="@string/158c1" />
1154     </TextView>
1155     ---
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_de_servicos.xml

```
1156         android:layout_margin="8dp"
1157         android:layout_weight="3"
1158         android:width="0dp"
1159         android:text="@string/158c2" />
1160     </TableRow>
1161
1162     <TableRow
1163         android:layout_width="match_parent"
1164         android:layout_height="wrap_content"
1165         android:background="#e0e0e0"
1166         android:gravity="center">
1167
1168         <TextView
1169             android:layout_margin="8dp"
1170             android:layout_weight="2"
1171             android:width="0dp"
1172             android:text="@string/159c1" />
1173
1174         <TextView
1175             android:layout_margin="8dp"
1176             android:layout_weight="3"
1177             android:width="0dp"
1178             android:text="@string/159c2" />
1179     </TableRow>
1180
1181     <TableRow
1182         android:layout_width="match_parent"
1183         android:layout_height="wrap_content"
1184         android:background="@color/primary_light"
1185         android:gravity="center">
1186
1187         <TextView
1188             android:layout_margin="8dp"
1189             android:layout_weight="2"
1190             android:width="0dp"
```



```
1191         android:text="@string/l60c1" />
1192
1193         <TextView
1194             android:layout_margin="8dp"
1195             android:layout_weight="3"
1196             android:width="0dp"
1197             android:text="@string/l60c2" />
1198     </TableRow>
1199
1200     <TableRow
1201         android:layout_width="match_parent"
1202         android:layout_height="wrap_content"
1203         android:background="#e0e0e0"
1204         android:gravity="center">
1205
1206         <TextView
1207             android:layout_margin="8dp"
1208             android:layout_weight="2"
1209             android:width="0dp"
1210             android:text="@string/l61c1" />
1211
1212         <TextView
1213             android:layout_margin="8dp"
1214             android:layout_weight="3"
1215             android:width="0dp"
1216             android:text="@string/l61c2" />
1217     </TableRow>
1218
1219     <TableRow
1220         android:layout_width="match_parent"
1221         android:layout_height="wrap_content"
1222         android:background="@color/primary_light"
1223         android:gravity="center">
1224         <TextView
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
1226         android:layout_margin="8dp"
1227         android:layout_weight="2"
1228         android:width="0dp"
1229         android:text="@string/l62c1" />
1230
1231     <TextView
1232         android:layout_margin="8dp"
1233         android:layout_weight="3"
1234         android:width="0dp"
1235         android:text="@string/l62c2" />
1236
1237 </TableRow>
1238
1239 <TableRow
1240     android:layout_width="match_parent"
1241     android:layout_height="wrap_content"
1242     android:background="#e0e0e0"
1243     android:gravity="center">
1244
1245     <TextView
1246         android:layout_margin="8dp"
1247         android:layout_weight="2"
1248         android:width="0dp"
1249         android:text="@string/l63c1" />
1250
1251     <TextView
1252         android:layout_margin="8dp"
1253         android:layout_weight="3"
1254         android:width="0dp"
1255         android:text="@string/l63c2" />
1256
1257 </TableRow>
1258
1259 <TableRow
1260     android:layout_width="match_parent"
1261     android:layout_height="wrap_content"
1262     android:background="@color/primary_light"
1263     ...
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
1261         android:gravity="center">
1262
1263         <TextView
1264             android:layout_margin="8dp"
1265             android:layout_weight="2"
1266             android:width="0dp"
1267             android:text="@string/l64c1" />
1268
1269         <TextView
1270             android:layout_margin="8dp"
1271             android:layout_weight="3"
1272             android:width="0dp"
1273             android:text="@string/l64c2" />
1274     </TableRow>
1275
1276
1277     <TableRow
1278         android:layout_width="match_parent"
1279         android:layout_height="wrap_content"
1280         android:background="#e0e0e0"
1281         android:gravity="center">
1282
1283         <TextView
1284             android:layout_margin="8dp"
1285             android:layout_weight="2"
1286             android:width="0dp"
1287             android:text="@string/l65c1" />
1288
1289         <TextView
1290             android:layout_margin="8dp"
1291             android:layout_weight="3"
1292             android:width="0dp"
1293             android:text="@string/l65c2" />
1294     </TableRow>
1295     -----
```

```
1296 <TableRow
1297     android:layout_width="match_parent"
1298     android:layout_height="wrap_content"
1299     android:background="@color/primary_light"
1300     android:gravity="center">
1301
1302     <TextView
1303         android:layout_margin="8dp"
1304         android:layout_weight="2"
1305         android:width="0dp"
1306         android:text="@string/l66c1" />
1307
1308     <TextView
1309         android:layout_margin="8dp"
1310         android:layout_weight="3"
1311         android:width="0dp"
1312         android:text="@string/l66c2" />
1313 </TableRow>
1314
1315 <TableRow
1316     android:layout_width="match_parent"
1317     android:layout_height="wrap_content"
1318     android:background="#e0e0e0"
1319     android:gravity="center">
1320
1321     <TextView
1322         android:layout_margin="8dp"
1323         android:layout_weight="2"
1324         android:width="0dp"
1325         android:text="@string/l67c1" />
1326
1327     <TextView
1328         android:layout_margin="8dp"
1329         android:layout_weight="3"
1330         android:width="0dp"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_de_servicos.xml

```
1331         android:text="@string/l67c2" />
1332     </TableRow>
1333
1334     </TableLayout>
1335
1336     </android.support.v4.widget.NestedScrollView>
1337
1338     </android.support.design.widget.CoordinatorLayout>
1339
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_estabelecimentos.xml

```
1 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:app="http://schemas.android.com/apk/res-auto"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent">
5
6   <android.support.design.widget.AppBarLayout
7     android:layout_width="match_parent"
8     android:layout_height="wrap_content"
9     android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
10
11     <android.support.v7.widget.Toolbar
12       android:id="@+id/toolbar_actvt_ajuda_tipos_estabelecimentos"
13       android:layout_width="match_parent"
14       android:layout_height="?attr/actionBarSize"
15       android:background="?attr/colorPrimary"
16       app:layout_scrollFlags="scroll|enterAlways"
17       app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
18   </android.support.design.widget.AppBarLayout>
19
20   <android.support.v4.widget.NestedScrollView
21     android:id="@+id/nestedScroll"
22     android:layout_width="match_parent"
23     android:layout_height="match_parent"
24     android:scrollbars="vertical"
25     app:layout_behavior="@string/appbar_scrolling_view_behavior">
26
27     <LinearLayout
28       android:layout_width="match_parent"
29       android:layout_height="match_parent"
30       android:layout_margin="16dp"
31       android:orientation="vertical"
32       android:paddingBottom="16dp">
33
34       <TextView
35         android:layout_width="match_parent"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_estabelecimentos.xml

```
36         android:layout_height="wrap_content"
37         android:text="@string/centro_hemato_titulo"
38         android:textAppearance="@android:style/TextAppearance.Medium" />
39
40     <TextView
41         android:layout_width="match_parent"
42         android:layout_height="wrap_content"
43         android:layout_marginBottom="16dp"
44         android:text="@string/centro_hemato" />
45
46     <TextView
47         android:layout_width="match_parent"
48         android:layout_height="wrap_content"
49         android:text="@string/caps_titulo"
50         android:textAppearance="@android:style/TextAppearance.Medium" />
51
52     <TextView
53         android:layout_width="match_parent"
54         android:layout_height="wrap_content"
55         android:layout_marginBottom="16dp"
56         android:text="@string/caps" />
57
58     <TextView
59         android:layout_width="match_parent"
60         android:layout_height="wrap_content"
61         android:text="@string/parto_isolado_titulo"
62         android:textAppearance="@android:style/TextAppearance.Medium" />
63
64     <TextView
65         android:layout_width="match_parent"
66         android:layout_height="wrap_content"
67         android:layout_marginBottom="16dp"
68         android:text="@string/parto_isolado" />
69
70     <TextView
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_estabelecimentos.xml

```
71         android:layout_width="match_parent"
72         android:layout_height="wrap_content"
73         android:text="@string/centro_saude_titulo"
74         android:textAppearance="@android:style/TextAppearance.Medium" />
75
76     <TextView
77         android:layout_width="match_parent"
78         android:layout_height="wrap_content"
79         android:layout_marginBottom="16dp"
80         android:text="@string/centro_saude" />
81
82     <TextView
83         android:layout_width="match_parent"
84         android:layout_height="wrap_content"
85         android:text="@string/clinica_titulo"
86         android:textAppearance="@android:style/TextAppearance.Medium" />
87
88     <TextView
89         android:layout_width="match_parent"
90         android:layout_height="wrap_content"
91         android:layout_marginBottom="16dp"
92         android:text="@string/clinica" />
93
94     <TextView
95         android:layout_width="match_parent"
96         android:layout_height="wrap_content"
97         android:text="@string/consultorio_isolado_titulo"
98         android:textAppearance="@android:style/TextAppearance.Medium" />
99
100    <TextView
101        android:layout_width="match_parent"
102        android:layout_height="wrap_content"
103        android:layout_marginBottom="16dp"
104        android:text="@string/consultorio_isolado" />
105    --
```


File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity ajuda_tipos_estabelecimentos.xml

```
106 <TextView
107     android:layout_width="match_parent"
108     android:layout_height="wrap_content"
109     android:text="@string/hospital_especializado_titulo"
110     android:textAppearance="@android:style/TextAppearance.Medium" />
111
112 <TextView
113     android:layout_width="match_parent"
114     android:layout_height="wrap_content"
115     android:layout_marginBottom="16dp"
116     android:text="@string/hospital_especializado" />
117
118 <TextView
119     android:layout_width="match_parent"
120     android:layout_height="wrap_content"
121     android:text="@string/hospital_geral_titulo"
122     android:textAppearance="@android:style/TextAppearance.Medium" />
123
124 <TextView
125     android:layout_width="match_parent"
126     android:layout_height="wrap_content"
127     android:layout_marginBottom="16dp"
128     android:text="@string/hospital_geral" />
129
130 <TextView
131     android:layout_width="match_parent"
132     android:layout_height="wrap_content"
133     android:text="@string/hospital_dia_titulo"
134     android:textAppearance="@android:style/TextAppearance.Medium" />
135
136 <TextView
137     android:layout_width="match_parent"
138     android:layout_height="wrap_content"
139     android:layout_marginBottom="16dp"
140     android:text="@string/hospital_dia" />
```

```

141 <TextView
142     android:layout_width="match_parent"
143     android:layout_height="wrap_content"
144     android:text="@string/labo_saude_publica_titulo"
145     android:textAppearance="@android:style/TextAppearance.Medium" />
146
147
148 <TextView
149     android:layout_width="match_parent"
150     android:layout_height="wrap_content"
151     android:layout_marginBottom="16dp"
152     android:text="@string/labo_saude_publica" />
153
154
155 <TextView
156     android:layout_width="match_parent"
157     android:layout_height="wrap_content"
158     android:text="@string/policlinica_titulo"
159     android:textAppearance="@android:style/TextAppearance.Medium" />
160
161
162 <TextView
163     android:layout_width="match_parent"
164     android:layout_height="wrap_content"
165     android:layout_marginBottom="16dp"
166     android:text="@string/policlinica" />
167
168
169 <TextView
170     android:layout_width="match_parent"
171     android:layout_height="wrap_content"
172     android:text="@string/posto_de_saude_titulo"
173     android:textAppearance="@android:style/TextAppearance.Medium" />
174
175
176 <TextView
177     android:layout_width="match_parent"
178     android:layout_height="wrap_content"
179     android:layout_marginBottom="16dp"

```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_ajuda_tipos_estabelecimentos.xml

```
176         android:text="@string/posto_de_saude" />
177
178     <TextView
179         android:layout_width="match_parent"
180         android:layout_height="wrap_content"
181         android:text="@string/upa_titulo"
182         android:textAppearance="@android:style/TextAppearance.Medium" />
183
184     <TextView
185         android:layout_width="match_parent"
186         android:layout_height="wrap_content"
187         android:layout_marginBottom="16dp"
188         android:text="@string/upa" />
189
190     <TextView
191         android:layout_width="match_parent"
192         android:layout_height="wrap_content"
193         android:text="@string/pronto_socorro_especializado_titulo"
194         android:textAppearance="@android:style/TextAppearance.Medium" />
195
196     <TextView
197         android:layout_width="match_parent"
198         android:layout_height="wrap_content"
199         android:layout_marginBottom="16dp"
200         android:text="@string/pronto_socorro_especializado" />
201
202     <TextView
203         android:layout_width="match_parent"
204         android:layout_height="wrap_content"
205         android:text="@string/pronto_socorro_geral_titulo"
206         android:textAppearance="@android:style/TextAppearance.Medium" />
207
208     <TextView
209         android:layout_width="match_parent"
210         android:layout_height="wrap_content"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_ajuda_tipos_estabelecimentos.xml

```
211         android:layout_marginBottom="16dp"
212         android:text="@string/pronto_socorro_geral" />
213
214     <TextView
215         android:layout_width="match_parent"
216         android:layout_height="wrap_content"
217         android:text="@string/sadt_titulo"
218         android:textAppearance="@android:style/TextAppearance.Medium" />
219
220     <TextView
221         android:layout_width="match_parent"
222         android:layout_height="wrap_content"
223         android:layout_marginBottom="16dp"
224         android:text="@string/sadt" />
225
226     <TextView
227         android:layout_width="match_parent"
228         android:layout_height="wrap_content"
229         android:text="@string/unidade_mista_titulo"
230         android:textAppearance="@android:style/TextAppearance.Medium" />
231
232     <TextView
233         android:layout_width="match_parent"
234         android:layout_height="wrap_content"
235         android:layout_marginBottom="16dp"
236         android:text="@string/unidade_mista" />
237
238     </LinearLayout>
239
240     </android.support.v4.widget.NestedScrollView>
241 </android.support.design.widget.CoordinatorLayout>
242 ~~~
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_busca_geral.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:fab="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/activity_busca_geral_layout"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     android:fitsSystemWindows="true"
10    tools:context="br.com.iucemar.dimon.cadesaude.BuscaGeralActivity">
11
12    <android.support.design.widget.AppBarLayout
13        android:layout_width="match_parent"
14        android:layout_height="wrap_content"
15        android:theme="@style/MeuTemaMaterial.AppBarOverlay">
16
17        <android.support.v7.widget.Toolbar
18            android:id="@+id/toolbar_actvt_busca_geral"
19            android:layout_width="match_parent"
20            android:layout_height="?attr/actionBarSize"
21            android:background="?attr/colorPrimary"
22            app:popupTheme="@style/MeuTemaMaterial.PopupOverlay" />
23
24    </android.support.design.widget.AppBarLayout>
25
26    <include layout="@layout/content_busca_geral" />
27
28    <com.github.clans.fab.FloatingActionButton
29        android:id="@+id/fab_activity_busca_geral"
30        android:layout_width="wrap_content"
31        android:layout_height="wrap_content"
32        android:layout_gravity="end|bottom"
33        android:layout_margin="16dp"
34        android:src="@drawable/ic_magnify_white_24dp"
35        fab:fab_label="Busca por tipos"
36    />
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_busca_geral.xml

```
36     fab:fab_size="normal" />
37
38 </android.support.design.widget.CoordinatorLayout>
39
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_busca_gps.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:fab="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/activity_busca_gps_layout"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     android:fitsSystemWindows="true"
10    tools:context="br.com.iucemar.dimon.cadesaude.BuscaGPSActivity">
11
12    <android.support.design.widget.AppBarLayout
13        android:layout_width="match_parent"
14        android:layout_height="wrap_content"
15        android:theme="@style/MeuTemaMaterial.AppBarOverlay">
16
17        <android.support.v7.widget.Toolbar
18            android:id="@+id/toolbar_actvt_busca_gps"
19            android:layout_width="match_parent"
20            android:layout_height="?attr/actionBarSize"
21            app:popupTheme="@style/MeuTemaMaterial.PopupOverlay" />
22
23    </android.support.design.widget.AppBarLayout>
24
25    <include layout="@layout/content_busca_gps" />
26
27    <com.github.clans.fab.FloatingActionButton
28        android:id="@+id/fab_filtro_menu"
29        android:layout_width="match_parent"
30        android:layout_height="match_parent"
31        android:layout_gravity="end|bottom"
32        android:elevation="2dp"
33        android:padding="@dimen/text_margin"
34        android:visibility="gone"
35        fab:menu_animationDelayPerItem="30"
```

```

36 fab:menu_backgroundColor="#AAFFFFFF"
37 fab:menu_buttonSpacing="0dp"
38 fab:menu_colorNormal="#E53935"
39 fab:menu_colorPressed="#FF5252"
40 fab:menu_colorRipple="#99FFFFFF"
41 fab:menu_fab_hide_animation="@anim/fab_scale_down"
42 fab:menu_fab_label="Fechar"
43 fab:menu_fab_show_animation="@anim/fab_scale_up"
44 fab:menu_fab_size="normal"
45 fab:menu_icon="@drawable/ic_filter_white_24dp"
46 fab:menu_labels_colorNormal="#333333"
47 fab:menu_labels_colorPressed="#444444"
48 fab:menu_labels_colorRipple="#66FFFFFF"
49 fab:menu_labels_cornerRadius="3dp"
50 fab:menu_labels_ellipsize="none"
51 fab:menu_labels_margin="0dp"
52 fab:menu_labels_maxLines="1"
53 fab:menu_labels_padding="8dp"
54 fab:menu_labels_paddingBottom="4dp"
55 fab:menu_labels_paddingLeft="8dp"
56 fab:menu_labels_paddingRight="8dp"
57 fab:menu_labels_paddingTop="4dp"
58 fab:menu_labels_position="left"
59 fab:menu_labels_showShadow="true"
60 fab:menu_labels_singleLine="false"
61 fab:menu_labels_textColor="#FFFFFF"
62 fab:menu_labels_textSize="14sp"
63 fab:menu_openDirection="up"
64 fab:menu_shadowColor="#60000000"
65 fab:menu_shadowRadius="2dp"
66 fab:menu_shadowXOffset="1dp"
67 fab:menu_shadowYOffset="2dp"

```

```
<com.github.clans.fab.FloatingActionButton
```


File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_busca_gps.xml

```
71     android:id="@+id/fab_filtro_limpar_todos"
72     android:layout_width="wrap_content"
73     android:layout_height="wrap_content"
74     android:src="@drawable/ic_filter_remove_white_24dp"
75     fab:fab_label="@string/fab_filtro_limpar_todos"
76     fab:fab_size="mini" />
77
78     <com.github.clans.fab.FloatingActionButton
79         android:id="@+id/fab_filtro_tipo_conven"
80         android:layout_width="wrap_content"
81         android:layout_height="wrap_content"
82         android:src="@drawable/ic_credit_card_white_24dp"
83         fab:fab_label="@string/fab_filtro_tipo_convenio"
84         fab:fab_size="mini" />
85
86     <com.github.clans.fab.FloatingActionButton
87         android:id="@+id/fab_filtro_tipo_atend"
88         android:layout_width="wrap_content"
89         android:layout_height="wrap_content"
90         android:src="@drawable/ic_healing_white_24dp"
91         fab:fab_label="@string/fab_filtro_tipo_atendimento"
92         fab:fab_size="mini" />
93
94     <com.github.clans.fab.FloatingActionButton
95         android:id="@+id/fab_filtro_tipo_prof"
96         android:layout_width="wrap_content"
97         android:layout_height="wrap_content"
98         android:src="@drawable/ic_school_white_24dp"
99         fab:fab_label="@string/fab_filtro_tipo_profissional"
100        fab:fab_size="mini" />
101
102     <com.github.clans.fab.FloatingActionButton
103         android:id="@+id/fab_filtro_tipo_serv"
104         android:layout_width="wrap_content"
105         android:layout_height="wrap_content"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_busca_gps.xml

```
106     android:src="@drawable/ic_local_hospital_white_24dp"
107     fab:fab_label="@string/fab_filtro_tipo_servico"
108     fab:fab_size="mini" />
109
110     <com.github.clans.fab.FloatingActionButton
111         android:id="@+id/fab_filtro_tipo_estab"
112         android:layout_width="wrap_content"
113         android:layout_height="wrap_content"
114         android:src="@drawable/ic_stethoscope_white_24dp"
115         fab:fab_label="@string/fab_filtro_tipo_estabelecimento"
116         fab:fab_size="mini" />
117     </com.github.clans.fab.FloatingActionButton>
118
119     </android.support.design.widget.CoordinatorLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_estabelecimento.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:fab="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:fitsSystemWindows="true"
9     tools:context="br.com.iucemar.dimon.cadesaude.EstabelecimentoActivity">
10
11     <android.support.design.widget.AppBarLayout
12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14         android:theme="@style/MeuTemaMaterial.AppBarOverlay">
15
16         <android.support.v7.widget.Toolbar
17             android:id="@+id/toolbar_actvt_estabelecimento"
18             android:layout_width="match_parent"
19             android:layout_height="wrap_content"
20             android:background="?attr/colorPrimary"
21             app:layout_scrollFlags="scroll|enterAlways"
22             app:popupTheme="@style/MeuTemaMaterial.PopupOverlay">
23
24             <TextView
25                 android:id="@+id/titulo_estab_activity_estab"
26                 android:layout_width="match_parent"
27                 android:layout_height="match_parent"
28                 android:layout_marginRight="8dp"
29                 android:gravity="center_vertical"
30                 android:textAppearance="@android:style/TextAppearance.Medium" />
31             </android.support.v7.widget.Toolbar>
32
33     <android.support.design.widget.TabLayout
34         android:id="@+id/tabs"
35         android:layout_width="match_parent"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_estabelecimento.xml

```
36     android:layout_height="wrap_content"
37     app:tabGravity="fill"
38     app:tabIndicatorColor="@color/icons"
39     app:tabMode="scrollable"
40     app:tabSelectedTextColor="@color/icons"
41     app:tabTextColor="@color/primary_light" />
42 </android.support.design.widget.AppBarLayout>
43
44 <ProgressBar
45     android:id="@+id/barra_progresso_estabelecimento_activity"
46     android:layout_width="wrap_content"
47     android:layout_height="wrap_content"
48     android:layout_gravity="center"
49     android:visibility="visible" />
50
51 <android.support.v4.view.ViewPager
52     android:id="@+id/viewpager"
53     android:layout_width="match_parent"
54     android:layout_height="match_parent"
55     android:background="@color/background_material_light"
56     android:visibility="gone"
57     app:layout_behavior="@string/appbar_scrolling_view_behavior" />
58 </android.support.design.widget.CoordinatorLayout>
59
60
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:fab="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/coordinator_layout_main_activity"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     android:fitsSystemWindows="true"
10    tools:context="br.com.iucemar.dimon.cadesaude.MainActivity">
11
12    <android.support.design.widget.AppBarLayout
13        android:id="@+id/appbar"
14        android:layout_width="match_parent"
15        android:layout_height="wrap_content"
16        android:theme="@style/MeuTemaMaterial.AppBarOverlay">
17
18        <android.support.v7.widget.Toolbar
19            android:id="@+id/toolbar_actvt_main"
20            android:layout_width="match_parent"
21            android:layout_height="?attr/actionBarSize"
22            android:background="?attr/colorPrimary"
23            app:popupTheme="@style/MeuTemaMaterial.PopupOverlay" />
24
25        <LinearLayout
26            android:layout_width="match_parent"
27            android:layout_height="wrap_content"
28            android:orientation="horizontal"
29            android:padding="16dp">
30
31            <ImageView
32                android:layout_width="wrap_content"
33                android:layout_height="wrap_content"
34                android:layout_marginRight="16dp"
35                android:src="@drawable/ic_star_white_24dp" />
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_main.xml

```
36
37     <TextView
38         android:layout_width="match_parent"
39         android:layout_height="wrap_content"
40         android:layout_gravity="center_vertical"
41         android:text="Meus favoritos"
42         android:textAppearance="@android:style/TextAppearance.Medium" />
43     </LinearLayout>
44 </android.support.design.widget.AppBarLayout>
45
46 <include layout="@layout/content_main" />
47
48 <com.github.clans.fab.FloatingActionButton
49     android:id="@+id/fab_busca_parametros"
50     android:layout_width="wrap_content"
51     android:layout_height="wrap_content"
52     android:layout_gravity="end|bottom"
53     android:layout_marginBottom="16dp"
54     android:layout_marginRight="16dp"
55     android:padding="@dimen/text_margin"
56     android:src="@drawable/ic_magnify_white_24dp"
57     app:fab_colorNormal="#DA4336"
58     app:fab_colorPressed="#E75043"
59     app:fab_colorRipple="#99FFFFFF"
60     app:fab_hideAnimation="@anim/fab_scale_down"
61     app:fab_label="Busca por parâmetros"
62     app:fab_shadowColor="#60000000"
63     app:fab_shadowRadius="4dp"
64     app:fab_shadowXOffset="1dp"
65     app:fab_shadowYOffset="3dp"
66     app:fab_showAnimation="@anim/fab_scale_up"
67     app:fab_showShadow="true"
68     xmlns:android="http://schemas.android.com/apk/res/android" />
69
70 <com.github.clans.fab.FloatingActionButton
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_main.xml

```
71     android:id="@+id/fab_busca_gps"
72     android:layout_width="wrap_content"
73     android:layout_height="wrap_content"
74     android:layout_marginBottom="88dp"
75     android:layout_marginRight="16dp"
76     android:padding="@dimen/text_margin"
77     android:src="@drawable/ic_crosshairs_gps_white_24dp"
78     app:fab_colorNormal="#DA4336"
79     app:fab_colorPressed="#E75043"
80     app:fab_colorRipple="#99FFFFFF"
81     app:fab_hideAnimation="@anim/fab_scale_down"
82     app:fab_label="Busca      por      localização"
83     app:fab_shadowColor="#60000000"
84     app:fab_shadowRadius="4dp"
85     app:fab_shadowXOffset="1dp"
86     app:fab_shadowYOffset="3dp"
87     app:fab_showAnimation="@anim/fab_scale_up"
88     app:fab_showShadow="true"
89     app:fab_size="normal"
90     app:layout_anchor="@id/fab_busca_parametros"
91     app:layout_anchorGravity="top|center" />
92 </android.support.design.widget.CoordinatorLayout>
93
94
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_primeira_configuracao.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2
3 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/layout_primeira_config_activity"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     tools:context="br.com.jucemar.dimon.cadesaude.PrimeiraConfiguracaoActivity">
10
11     <LinearLayout
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:layout_gravity="center"
15         android:orientation="vertical">
16
17         <ProgressBar
18             android:id="@+id/barra_progresso_primeira_conf_activity"
19             android:layout_width="wrap_content"
20             android:layout_height="wrap_content"
21             android:layout_gravity="center"
22             android:visibility="visible" />
23
24         <ImageView
25             android:id="@+id/background_erro_primeira_conf_activity"
26             android:layout_width="wrap_content"
27             android:layout_height="wrap_content"
28             android:layout_gravity="center_horizontal"
29             android:src="@drawable/crosshairs"
30             android:visibility="gone" />
31
32         <TextView
33             android:id="@+id/label_erro_primeira_config_activity"
34             android:layout_width="wrap_content"
35             android:layout_height="wrap_content"
```


File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_primeira_configuracao.xml

```
36     android:layout_gravity="center_horizontal"
37     android:layout_marginTop="30dp"
38     android:text="@string/txt_erro_gps"
39     android:textAppearance="@android:style/TextAppearance.Large"
40     android:textColor="#BDBDBD"
41     android:visibility="gone" />
42     </LinearLayout>
43
44 </FrameLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_primeiros_passos.xml

```
1 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   tools:context="br.com.jucemar_dimon.cadesaude.PrimeirosPassosActivity">
6
7   <LinearLayout
8       android:layout_width="match_parent"
9       android:layout_height="match_parent"
10      android:background="@color/primary_light"
11      android:orientation="vertical">
12
13       <ImageView
14           android:layout_width="wrap_content"
15           android:layout_height="0dp"
16           android:layout_marginBottom="16dp"
17           android:layout_marginTop="72dp"
18           android:layout_weight="2"
19           android:src="@drawable/ic_launcher_web" />
20
21       <TextView
22           android:id="@+id/fullscreen_content"
23           android:layout_width="wrap_content"
24           android:layout_height="0dp"
25           android:layout_margin="16dp"
26           android:layout_marginBottom="16dp"
27           android:layout_weight="2"
28           android:gravity="center_horizontal"
29           android:keepScreenOn="true"
30           android:text="@string/dummy_content"
31           android:textColor="@color/primary_text"
32           android:textSize="14sp" />
33
34   </LinearLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_primeiros_passos.xml

```
36 <FrameLayout
37     android:layout_width="match_parent"
38     android:layout_height="match_parent"
39     android:layout_gravity="left|center_vertical"
40     android:fitsSystemWindows="true">
41
42     <LinearLayout
43         android:id="@+id/fullscreen_content_controls"
44         style="?metaButtonBarStyle"
45         android:layout_width="match_parent"
46         android:layout_height="wrap_content"
47         android:layout_gravity="bottom|center_horizontal"
48         android:background="@color/primary"
49         android:orientation="horizontal"
50         tools:ignore="UselessParent">
51
52         <Button
53             android:id="@+id/dummy_button"
54             style="?metaButtonBarButtonStyle"
55             android:layout_width="0dp"
56             android:layout_height="wrap_content"
57             android:layout_weight="1"
58             android:enabled="true"
59             android:text="@string/dummy_button"
60             android:textAppearance="@android:style/TextAppearance.Medium"
61             android:textColor="@color/primary_light" />
62
63     </LinearLayout>
64 </FrameLayout>
65
66 </FrameLayout>
67 ~
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_sobre.xml

```
1 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:app="http://schemas.android.com/apk/res-auto"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent">
5
6   <android.support.design.widget.AppBarLayout
7     android:layout_width="match_parent"
8     android:layout_height="wrap_content"
9     android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
10
11     <android.support.v7.widget.Toolbar
12       android:id="@+id/toolbar_actvt_sobre"
13       android:layout_width="match_parent"
14       android:layout_height="?attr/actionBarSize"
15       android:background="?attr/colorPrimary"
16       app:layout_scrollFlags="scroll|enterAlways"
17       app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
18
19   </android.support.design.widget.AppBarLayout>
20
21   <android.support.v4.widget.NestedScrollView
22     android:id="@+id/nestedScroll"
23     android:layout_width="match_parent"
24     android:layout_height="match_parent"
25     android:scrollbars="vertical"
26     app:layout_behavior="@string/appbar_scrolling_view_behavior">
27
28     <LinearLayout
29       android:layout_width="match_parent"
30       android:layout_height="wrap_content"
31       android:layout_margin="16dp"
32       android:orientation="vertical">
33
34       <TextView
35         android:layout_width="match_parent"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_sobre.xml

```
36     android:layout_height="match_parent"
37     android:layout_marginBottom="16dp"
38     android:layout_marginTop="16dp"
39     android:gravity="center_horizontal"
40     android:text="@string/appNome" />
41
42     <ImageView
43         android:layout_width="wrap_content"
44         android:layout_height="wrap_content"
45         android:layout_gravity="center_horizontal"
46         android:src="@mipmap/ic_launcher" />
47
48     <TextView
49         android:id="@+id/sobre_ttv_app_versao"
50         android:layout_width="match_parent"
51         android:layout_height="match_parent"
52         android:layout_marginTop="16dp"
53         android:gravity="center_horizontal"
54         android:text="@string/appVersao" />
55
56     <TextView
57         android:id="@+id/sobre_ttv_bd_versao"
58         android:layout_width="match_parent"
59         android:layout_height="match_parent"
60         android:layout_marginTop="16dp"
61         android:gravity="center_horizontal"
62         android:text="@string/bdVersao" />
63
64     <TextView
65         android:layout_width="match_parent"
66         android:layout_height="match_parent"
67         android:layout_gravity="fill_horizontal"
68         android:layout_marginTop="16dp"
69         android:gravity="center_horizontal"
70         android:paddingBottom="32dp"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\activity_sobre.xml

```
71         android:text="@string/appDesc" />
72
73         </LinearLayout>
74         </android.support.v4.widget.NestedScrollView>
75
76     </android.support.design.widget.CoordinatorLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_tela_inicial.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2
3 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:background="@drawable/city"
8     android:orientation="vertical"
9     android:paddingBottom="@dimen/activity_vertical_margin"
10    android:paddingLeft="@dimen/activity_horizontal_margin"
11    android:paddingRight="@dimen/activity_horizontal_margin"
12    android:paddingTop="@dimen/activity_vertical_margin"
13    tools:context="br.com.iucemar.dimon.cadesaude.TelaInicialActivity">
14
15    <ImageView
16        android:id="@+id/imageView"
17        android:layout_width="wrap_content"
18        android:layout_height="0dp"
19        android:layout_gravity="center_vertical"
20        android:layout_marginTop="64dp"
21        android:layout_weight="2"
22        android:src="@drawable/ic_launcher_web" />
23
24    <TextView
25        android:layout_width="wrap_content"
26        android:layout_height="0dp"
27        android:layout_gravity="center_horizontal"
28        android:layout_weight="1"
29        android:gravity="center"
30        android:text="CadEsaúde"
31        android:textAppearance="@android:style/TextAppearance.Large" />
32
33    <TextView
34        android:layout_width="wrap_content"
35        android:layout_height="0dp"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\activity_tela_inicial.xml

```
36     android:layout_gravity="fill_horizontal"
37     android:layout_marginBottom="0dp"
38     android:layout_weight="1"
39     android:gravity="center"
40     android:text="Cadastro dos Estabelecimentos de Saúde de Santa Catarina"
41     android:textAppearance="@android:style/TextAppearance.Medium" />
42
43     <TextView
44         android:id="@+id/txv_versao_app"
45         android:layout_width="wrap_content"
46         android:layout_height="0dp"
47         android:layout_gravity="right"
48         android:layout_weight="1"
49         android:gravity="bottom"
50         android:textAppearance="@android:style/TextAppearance.Small" />
51
52     </LinearLayout>
```

54

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content_busca_geral.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     app:layout_behavior="@string/appbar_scrolling_view_behavior"
8     tools:context="br.com.iucemar.dimon.cadesaude.MainActivity">
9
10    <android.support.v7.widget.RecyclerView
11        android:id="@+id/lista_estabelecimentos"
12        android:layout_width="match_parent"
13        android:layout_height="match_parent"
14        android:backgroundTint="@color/divider"
15        android:scrollbars="vertical"
16        android:visibility="gone" />
17
18    <ProgressBar
19        android:id="@+id/barra_progresso_busca_geral_activity"
20        android:layout_width="wrap_content"
21        android:layout_height="wrap_content"
22        android:layout_gravity="center"
23        android:visibility="visible" />
24
25    <LinearLayout
26        android:layout_width="wrap_content"
27        android:layout_height="wrap_content"
28        android:layout_gravity="center"
29        android:orientation="vertical">
30
31        <ImageView
32            android:id="@+id/background_erro_busca_geral"
33            android:layout_width="wrap_content"
34            android:layout_height="wrap_content"
35            android:layout_gravity="center_horizontal"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content_busca_geral.xml

```
36     android:src="@drawable/magnify_background"
37     android:visibility="visible" />
38
39     <TextView
40         android:id="@+id/label_erro_busca"
41         android:layout_width="wrap_content"
42         android:layout_height="wrap_content"
43         android:layout_gravity="center_horizontal"
44         android:text="@string/txt_erro_busca_geral"
45         android:textAppearance="@android:style/TextAppearance.Large"
46         android:textColor="#BDBDBD"
47         android:visibility="visible" />
48
49     <TextView
50         android:id="@+id/label_municipios_para_busca"
51         android:layout_width="wrap_content"
52         android:layout_height="wrap_content"
53         android:layout_gravity="center_horizontal"
54         android:layout_margin="16dp"
55         android:gravity="center"
56         android:textAppearance="@android:style/TextAppearance.Small"
57         android:textColor="#BDBDBD"
58         android:visibility="visible" />
59
60     </LinearLayout>
61
62 </FrameLayout>
63
64
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content_busca_gps.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     app:layout_behavior="@string/appbar_scrolling_view_behavior"
8     tools:context="br.com.iucemar.dimon.cadesaude.BuscaGPSActivity">
9
10    <android.support.v7.widget.RecyclerView
11        android:id="@+id/lista_estabelecimentos"
12        android:layout_width="match_parent"
13        android:layout_height="match_parent"
14        android:backgroundTint="@color/divider"
15        android:scrollbars="vertical"
16        android:visibility="gone" />
17
18    <LinearLayout
19        android:layout_width="wrap_content"
20        android:layout_height="wrap_content"
21        android:layout_gravity="center"
22        android:orientation="vertical">
23
24        <ProgressBar
25            android:id="@+id/barra_progresso_gps_activity"
26            android:layout_width="wrap_content"
27            android:layout_height="wrap_content"
28            android:layout_gravity="center"
29            android:visibility="visible" />
30
31    </LinearLayout>
32
33    <LinearLayout
34        android:layout_width="wrap_content"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content_busca_gps.xml

```
36     android:layout_height="wrap_content"
37     android:layout_gravity="center"
38     android:orientation="vertical">
39
40     <ImageView
41         android:id="@+id/background_erro_gps"
42         android:layout_width="wrap_content"
43         android:layout_height="wrap_content"
44         android:layout_gravity="center_horizontal"
45         android:src="@drawable/crosshairs"
46         android:visibility="gone" />
47
48     <TextView
49         android:id="@+id/label_erro_gps"
50         android:layout_width="wrap_content"
51         android:layout_height="wrap_content"
52         android:layout_gravity="center_horizontal"
53         android:layout_marginTop="30dp"
54         android:text="@string/txt_erro_gps"
55         android:textAppearance="@android:style/TextAppearance.Large"
56         android:textColor="#BDBDBD"
57         android:visibility="gone" />
58
59     </LinearLayout>
60 </FrameLayout>
61
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content_fragment_estabelecimento.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/scrool_estabelecimento"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent">
6
7     <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
8         xmlns:tools="http://schemas.android.com/tools"
9         android:layout_width="match_parent"
10        android:layout_height="match_parent"
11        android:padding="@dimen/text_margin">
12
13        <ImageButton
14            android:id="@+id/icone_razao_social"
15            android:layout_width="wrap_content"
16            android:layout_height="wrap_content"
17            android:layout_marginBottom="16dp"
18            android:layout_marginRight="16dp"
19            android:background="@android:color/transparent"
20            android:src="@drawable/ic_briefcase_grey600_24dp" />
21
22        <TextView
23            android:id="@+id/label_razao_social_abrir_estabelecimento"
24            android:layout_width="wrap_content"
25            android:layout_height="wrap_content"
26            android:layout_alignParentEnd="true"
27            android:layout_alignParentRight="true"
28            android:layout_alignTop="@+id/icone_razao_social"
29            android:layout_marginBottom="4dp"
30            android:layout_toEndOf="@+id/icone_razao_social"
31            android:layout_toRightOf="@+id/icone_razao_social"
32            android:layout="@id/icone_razao_social"
33            android:lines="2"         android:text="Razão Social:"
34            android:textAppearance="?android:attr/textAppearanceMedium" />
```

```

36
37
38     android:id="@+id/label_nome_fantasia_abrir_estabelecimento"
39     android:layout_width="wrap_content"
40     android:layout_height="wrap_content"
41     android:layout_alignLeft="@+id/label_razao_social_abrir_estabelecimento"
42     android:layout_alignParentRight="true"
43     android:layout_alignStart="@+id/label_razao_social_abrir_estabelecimento"
44     android:layout_below="@+id/label_razao_social_abrir_estabelecimento"
45     android:layout_marginBottom="16dp"
46     android:layout="@id/icone_razao_social"
47     android:lines="2"
48     android:text="Nome
49     Fantasia:"
50     android:textAppearance="?android:attr/textAppearanceMedium" />
51
52 <ImageButton
53     android:id="@+id/icone_tipo_estabelecimento"
54     android:layout_width="wrap_content"
55     android:layout_height="wrap_content"
56     android:layout_alignLeft="@+id/icone_razao_social"
57     android:layout_alignParentLeft="true"
58     android:layout_alignStart="@+id/icone_razao_social"
59     android:layout_below="@+id/label_nome_fantasia_abrir_estabelecimento"
60     android:layout_marginRight="16dp"
61     android:background="@android:color/transparent"
62     android:src="@drawable/ic_stethoscope_grev600_24dp" />
63
64 <TextView
65     android:id="@+id/label_tipo_estabelecimento_abrir_estabelecimento"
66     android:layout_width="wrap_content"
67     android:layout_height="wrap_content"
68     android:layout_alignParentRight="true"
69     android:layout_below="@+id/label_nome_fantasia_abrir_estabelecimento"
70     android:layout_marginBottom="16dp"
71     android:layout_toRightOf="@id/icone_tipo_estabelecimento"

```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content_fragment_estabelecimento.xml

```
71         android:background="@android:color/transparent"
72         android:text="Tipo de estabelecimento:"
73         android:textAppearance="?android:attr/textAppearanceMedium" />
74
75     <ImageButton
76         android:id="@+id/icone_endereco"
77         android:layout_width="wrap_content"
78         android:layout_height="wrap_content"
79         android:layout_alignParentLeft="true"
80         android:layout_below="@id/label_tipo_estabelecimento_abrir_estabelecimento"
81         android:layout_marginRight="16dp"
82         android:background="@android:color/transparent"
83         android:src="@drawable/ic_map_marker_grey600_24dp" />
84
85     <TextView
86         android:id="@+id/label_logradouro_abrir_estabelecimento"
87         android:layout_width="wrap_content"
88         android:layout_height="wrap_content"
89         android:layout_alignLeft="@+id/label_complemento_abrir_estabelecimento"
90         android:layout_alignParentRight="true"
91         android:layout_alignStart="@+id/label_complemento_abrir_estabelecimento"
92         android:layout_alignTop="@+id/icone_endereco"
93         android:layout_marginBottom="4dp"
94         android:text="Logradouro: "
95         android:textAppearance="?android:attr/textAppearanceMedium" />
96
97     <TextView
98         android:id="@+id/label_numero_abrir_estabelecimento"
99         android:layout_width="wrap_content"
100        android:layout_height="wrap_content"
101        android:layout_alignLeft="@+id/label_logradouro_abrir_estabelecimento"
102        android:layout_alignParentRight="true"
103        android:layout_alignStart="@+id/label_logradouro_abrir_estabelecimento"
104        android:layout_below="@+id/label_logradouro_abrir_estabelecimento"
105        android:layout_marginBottom="4dp"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\content_fragment_estabelecimento.xml

```
106         android:text="@+id/label_complemento_abrir_estabelecimento" />
107         android:textAppearance="?android:attr/textAppearanceMedium" />
108
109     <TextView
110         android:id="@+id/label_complemento_abrir_estabelecimento"
111         android:layout_width="wrap_content"
112         android:layout_height="wrap_content"
113         android:layout_alignLeft="@+id/label_municipio_abrir_estabelecimento"
114         android:layout_alignParentRight="true"
115         android:layout_alignStart="@+id/label_municipio_abrir_estabelecimento"
116         android:layout_below="@+id/label_numero_abrir_estabelecimento"
117         android:layout_marginBottom="4dp"
118         android:text="Complemento:"
119         android:textAppearance="?android:attr/textAppearanceMedium" />
120
121     <TextView
122         android:id="@+id/label_bairro_abrir_estabelecimento"
123         android:layout_width="wrap_content"
124         android:layout_height="wrap_content"
125         android:layout_alignLeft="@+id/label_complemento_abrir_estabelecimento"
126         android:layout_alignParentRight="true"
127         android:layout_alignStart="@+id/label_complemento_abrir_estabelecimento"
128         android:layout_below="@+id/label_complemento_abrir_estabelecimento"
129         android:layout_marginBottom="4dp"
130         android:text="Bairro:"
131         android:textAppearance="?android:attr/textAppearanceMedium" />
132
133     <TextView
134         android:id="@+id/label_cep_abrir_estabelecimento"
135         android:layout_width="wrap_content"
136         android:layout_height="wrap_content"
137         android:layout_alignLeft="@+id/label_bairro_abrir_estabelecimento"
138         android:layout_alignParentRight="true"
139         android:layout_alignStart="@+id/label_bairro_abrir_estabelecimento"
140         android:layout_below="@+id/label_bairro_abrir_estabelecimento"
```


File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\content_fragment_estabelecimento.xml

```
141         android:layout_marginBottom="4dp"
142         android:text="CEP:"
143         android:textAppearance="?android:attr/textAppearanceMedium" />
144
145     <TextView
146         android:id="@+id/label_municipio_abrir_estabelecimento"
147         android:layout_width="wrap_content"
148         android:layout_height="wrap_content"
149         android:layout_alignParentRight="true"
150         android:layout_below="@+id/label_cep_abrir_estabelecimento"
151         android:layout_marginBottom="4dp"
152         android:layout_toEndOf="@+id/icone_endereco"
153         android:layout_toRightOf="@+id/icone_endereco"
154         android:text="Município:"
155         android:textAppearance="?android:attr/textAppearanceMedium" />
156
157     <TextView
158         android:id="@+id/label_estado_abrir_estabelecimento"
159         android:layout_width="wrap_content"
160         android:layout_height="wrap_content"
161         android:layout_alignParentRight="true"
162         android:layout_below="@+id/label_municipio_abrir_estabelecimento"
163         android:layout_marginBottom="16dp"
164         android:layout_toEndOf="@+id/icone_endereco"
165         android:layout_toRightOf="@+id/icone_endereco"
166         android:text="Estado:"
167         android:textAppearance="?android:attr/textAppearanceMedium" />
168
169     <ImageButton
170         android:id="@+id/icone_telefone"
171         android:layout_width="wrap_content"
172         android:layout_height="wrap_content"
173         android:layout_alignParentLeft="true"
174         android:layout_below="@id/label_estado_abrir_estabelecimento"
175         android:layout_marginRight="16dp"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content_fragment_estabelecimento.xml

```
176         android:background="@android:color/transparent"
177         android:paddingBottom="16dp"
178         android:src="@drawable/ic_phone_grey600_24dp" />
179
180     <TextView
181         android:id="@+id/label_telefone_abrir_estabelecimento"
182         android:layout_width="wrap_content"
183         android:layout_height="wrap_content"
184         android:layout_alignBottom="@id/icone_telefone"
185         android:layout_below="@id/label_estado_abrir_estabelecimento"
186         android:layout_toEndOf="@+id/icone_telefone"
187         android:layout_toRightOf="@+id/icone_telefone"
188         android:text="Telefone:"
189         android:textAppearance="?android:attr/textAppearanceMedium" />
190
191     </RelativeLayout>
192 </ScrollView>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     app:layout_behavior="@string/appbar_scrolling_view_behavior"
8     tools:context="br.com.jucemar_dimon.cadesaude.MainActivity"
9     tools:showIn="@layout/activity_main">
10
11     <android.support.v7.widget.RecyclerView
12         android:id="@+id/lista_estabelecimentos"
13         android:layout_width="match_parent"
14         android:layout_height="match_parent"
15         android:backgroundTint="@color/divider"
16         android:scrollbars="vertical"
17         android:visibility="gone" />
18
19     <ProgressBar
20         android:id="@+id/barra_progresso_main_activity"
21         android:layout_width="wrap_content"
22         android:layout_height="wrap_content"
23         android:layout_gravity="center"
24         android:visibility="visible" />
25
26     <LinearLayout
27         android:layout_width="wrap_content"
28         android:layout_height="wrap_content"
29         android:layout_gravity="center"
30         android:orientation="vertical">
31
32         <ImageView
33             android:id="@+id/background_lista_vazia"
34             android:layout_width="wrap_content"
35             android:layout_height="wrap_content"
36         -
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\content_main.xml

```
36     android:layout_gravity="center_horizontal"
37     android:src="@drawable/stethoscope_background"
38     android:visibility="gone" />
39
40     <TextView
41         android:id="@+id/label_lista_sem_favoritos"
42         android:layout_width="wrap_content"
43         android:layout_height="wrap_content"
44         android:layout_gravity="center_horizontal"
45         android:layout_marginTop="30dp"
46         android:text="@string/txt_lista_sem_favoritos"
47         android:textAppearance="@android:style/TextAppearance.Large"
48         android:textColor="#BDBDBD"
49         android:visibility="gone" />
50
51     </LinearLayout>
52 </FrameLayout>
53
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\fragment_estabelecimento.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:fab="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/asaaas"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     android:fitsSystemWindows="false">
10
11     <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
12         android:id="@+id/scrool_estabelecimento"
13         android:layout_width="match_parent"
14         android:layout_height="match_parent"
15         android:paddingBottom="64dp">
16
17         <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
18             xmlns:tools="http://schemas.android.com/tools"
19             android:layout_width="match_parent"
20             android:layout_height="match_parent"
21             android:padding="@dimen/text_margin">
22
23             <ImageButton
24                 android:id="@+id/icone_razao_social"
25                 android:layout_width="wrap_content"
26                 android:layout_height="wrap_content"
27                 android:layout_marginBottom="16dp"
28                 android:layout_marginRight="16dp"
29                 android:background="@android:color/transparent"
30                 android:src="@drawable/ic_briefcase_grey600_24dp" />
31
32             <TextView
33                 android:id="@+id/label_razao_social_abrir_estabelecimento"
34                 android:layout_width="wrap_content"
35                 android:layout_height="wrap_content"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\fragment_estabelecimento.xml

```
36         android:layout_alignParentEnd="true"
37         android:layout_alignParentRight="true"
38         android:layout_alignTop="@+id/icone_razao_social"
39         android:layout_marginBottom="4dp"
40         android:layout_toEndOf="@+id/icone_razao_social"
41         android:layout_toRightOf="@+id/icone_razao_social"
42         android:layout="@id/icone_razao_social"
43         android:lines="2"
44         android:text="Razão"
45         android:textAppearance="?android:attr/textAppearanceMedium" />
46
47     <TextView
48         android:id="@+id/label_nome_fantasia_abrir_estabelecimento"
49         android:layout_width="wrap_content"
50         android:layout_height="wrap_content"
51         android:layout_alignLeft="@+id/label_razao_social_abrir_estabelecimento"
52         android:layout_alignParentRight="true"
53         android:layout_alignStart="@+id/label_razao_social_abrir_estabelecimento"
54         android:layout_below="@+id/label_razao_social_abrir_estabelecimento"
55         android:layout_marginBottom="16dp"
56         android:layout="@id/icone_razao_social"
57         android:lines="2"
58         android:text="Nome"
59         android:textAppearance="?android:attr/textAppearanceMedium" />
60
61     <ImageButton
62         android:id="@+id/icone_tipo_estabelecimento"
63         android:layout_width="wrap_content"
64         android:layout_height="wrap_content"
65         android:layout_alignLeft="@+id/icone_razao_social"
66         android:layout_alignParentLeft="true"
67         android:layout_alignStart="@+id/icone_razao_social"
68         android:layout_below="@+id/label_nome_fantasia_abrir_estabelecimento"
69         android:layout_marginRight="16dp"
70         android:background="@android:color/transparent"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\fragment_estabelecimento.xml

```
71         android:src="@drawable/ic_stethoscope_grey600_24dp" />
72
73     <TextView
74         android:id="@+id/label_tipo_estabelecimento_abrir_estabelecimento"
75         android:layout_width="wrap_content"
76         android:layout_height="wrap_content"
77         android:layout_alignParentRight="true"
78         android:layout_below="@+id/label_nome_fantasia_abrir_estabelecimento"
79         android:layout_marginBottom="16dp"
80         android:layout_toRightOf="@id/icone_tipo_estabelecimento"
81         android:background="@android:color/transparent"
82         android:text="Tipo" de estabelecimento:"
83         android:textAppearance="?android:attr/textAppearanceMedium" />
84
85     <ImageButton
86         android:id="@+id/icone_endereco"
87         android:layout_width="wrap_content"
88         android:layout_height="wrap_content"
89         android:layout_alignParentLeft="true"
90         android:layout_below="@id/label_tipo_estabelecimento_abrir_estabelecimento"
91         android:layout_marginRight="16dp"
92         android:background="@android:color/transparent"
93         android:src="@drawable/ic_map_marker_grey600_24dp" />
94
95     <TextView
96         android:id="@+id/label_logradouro_abrir_estabelecimento"
97         android:layout_width="wrap_content"
98         android:layout_height="wrap_content"
99         android:layout_alignLeft="@+id/label_complemento_abrir_estabelecimento"
100        android:layout_alignParentRight="true"
101        android:layout_alignStart="@+id/label_complemento_abrir_estabelecimento"
102        android:layout_alignTop="@+id/icone_endereco"
103        android:layout_marginBottom="4dp"
104        android:text="Logradouro:"
105        android:textAppearance="?android:attr/textAppearanceMedium" />
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\fragment_estabelecimento.xml

```
106
107
108         android:id="@+id/label_numero_abrir_estabelecimento"
109         android:layout_width="wrap_content"
110         android:layout_height="wrap_content"
111         android:layout_alignLeft="@+id/label_logradouro_abrir_estabelecimento"
112         android:layout_alignParentRight="true"
113         android:layout_alignStart="@+id/label_logradouro_abrir_estabelecimento"
114         android:layout_below="@+id/label_logradouro_abrir_estabelecimento"
115         android:layout_marginBottom="4dp"
116         android:text="Número:"
117         android:textAppearance="?android:attr/textAppearanceMedium" />
118
119     <TextView
120         android:id="@+id/label_complemento_abrir_estabelecimento"
121         android:layout_width="wrap_content"
122         android:layout_height="wrap_content"
123         android:layout_alignLeft="@+id/label_municipio_abrir_estabelecimento"
124         android:layout_alignParentRight="true"
125         android:layout_alignStart="@+id/label_municipio_abrir_estabelecimento"
126         android:layout_below="@+id/label_numero_abrir_estabelecimento"
127         android:layout_marginBottom="4dp"
128         android:text="Complemento:"
129         android:textAppearance="?android:attr/textAppearanceMedium" />
130
131     <TextView
132         android:id="@+id/label_bairro_abrir_estabelecimento"
133         android:layout_width="wrap_content"
134         android:layout_height="wrap_content"
135         android:layout_alignLeft="@+id/label_complemento_abrir_estabelecimento"
136         android:layout_alignParentRight="true"
137         android:layout_alignStart="@+id/label_complemento_abrir_estabelecimento"
138         android:layout_below="@+id/label_complemento_abrir_estabelecimento"
139         android:layout_marginBottom="4dp"
140         android:text="Bairro:"
```


File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\fragment_estabelecimento.xml

```
141         android:textAppearance="?android:attr/textAppearanceMedium" />
142
143     <TextView
144         android:id="@+id/label_cep_abrir_estabelecimento"
145         android:layout_width="wrap_content"
146         android:layout_height="wrap_content"
147         android:layout_alignLeft="@+id/label_bairro_abrir_estabelecimento"
148         android:layout_alignParentRight="true"
149         android:layout_alignStart="@+id/label_bairro_abrir_estabelecimento"
150         android:layout_below="@+id/label_bairro_abrir_estabelecimento"
151         android:layout_marginBottom="4dp"
152         android:text="CEP:"
153         android:textAppearance="?android:attr/textAppearanceMedium" />
154
155     <TextView
156         android:id="@+id/label_municipio_abrir_estabelecimento"
157         android:layout_width="wrap_content"
158         android:layout_height="wrap_content"
159         android:layout_alignParentRight="true"
160         android:layout_below="@+id/label_cep_abrir_estabelecimento"
161         android:layout_marginBottom="4dp"
162         android:layout_toEndOf="@+id/icone_endereco"
163         android:layout_toRightOf="@+id/icone_endereco"
164         android:text="Município:"
165         android:textAppearance="?android:attr/textAppearanceMedium" />
166
167     <TextView
168         android:id="@+id/label_estado_abrir_estabelecimento"
169         android:layout_width="wrap_content"
170         android:layout_height="wrap_content"
171         android:layout_alignParentRight="true"
172         android:layout_below="@+id/label_municipio_abrir_estabelecimento"
173         android:layout_marginBottom="16dp"
174         android:layout_toEndOf="@+id/icone_endereco"
175         android:layout_toRightOf="@+id/icone_endereco"
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\fragment_estabelecimento.xml

```
176         android:text="@Estado:"
177         android:textAppearance="?android:attr/textAppearanceMedium" />
178
179     <ImageButton
180         android:id="@+id/icone_telefone"
181         android:layout_width="wrap_content"
182         android:layout_height="wrap_content"
183         android:layout_alignParentLeft="true"
184         android:layout_below="@id/label_estado_abrir_estabelecimento"
185         android:layout_marginRight="16dp"
186         android:background="@android:color/transparent"
187         android:paddingBottom="16dp"
188         android:src="@drawable/ic_phone_grey600_24dp" />
189
190     <TextView
191         android:id="@+id/label_telefone_abrir_estabelecimento"
192         android:layout_width="wrap_content"
193         android:layout_height="wrap_content"
194         android:layout_alignBottom="@id/icone_telefone"
195         android:layout_below="@id/label_estado_abrir_estabelecimento"
196         android:layout_toEndOf="@+id/icone_telefone"
197         android:layout_toRightOf="@+id/icone_telefone"
198         android:text="Telefone:"
199         android:textAppearance="?android:attr/textAppearanceMedium" />
200
201     </RelativeLayout>
202 </ScrollView>
203
204 <com.github.clans.fab.FloatingActionButton
205     android:id="@+id/fab_add_favorito"
206     android:layout_width="wrap_content"
207     android:layout_height="wrap_content"
208     android:layout_gravity="right|bottom"
209     android:layout_marginBottom="72dp"
210     android:layout_marginRight="16dp"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\fragment_estabelecimento.xml

```
211     android:src="@drawable/ic_star_outline_white_24dp"
212     app:fab_colorNormal="#DA4336"
213     app:fab_colorPressed="#E75043"
214     app:fab_colorRipple="#99FFFFFF"
215     app:fab_hideAnimation="@anim/fab_scale_down"
216     app:fab_label="Busca por parâmetros"
217     app:fab_shadowColor="#66000000"
218     app:fab_shadowRadius="4dp"
219     app:fab_shadowXOffset="1dp"
220     app:fab_shadowYOffset="3dp"
221     app:fab_showAnimation="@anim/fab_scale_up"
222     app:fab_showShadow="true"
223     ann:fab_size="normal"/>
224
225 </android.support.design.widget.CoordinatorLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\fragment_listas.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/view_vazio"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:layout_marginBottom="50dp"
7     android:orientation="vertical">
8
9     <ListView
10         android:id="@+id/list_view"
11         android:layout_width="match_parent"
12         android:layout_height="match_parent"
13         android:layout_marginBottom="50dp"
14         android:listSelector="@android:color/transparent" />
15
16 </LinearLayout>
17
18
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\item_lista_estabelecimentos.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     android:layout_marginTop="2dp"
6     android:background="@drawable/selector"
7     android:orientation="vertical"
8     android:paddingBottom="8dp"
9     android:paddingLeft="16dp"
10    android:paddingRight="8dp"
11    android:paddingTop="8dp">
12
13    <TextView
14        android:id="@+id/nome_fantasia_lista_busca"
15        android:layout_width="match_parent"
16        android:layout_height="wrap_content"
17        android:layout_alignParentLeft="true"
18        android:layout_alignParentStart="true"
19        android:layout_marginBottom="5dp"
20        android:layout_toLeftOf="@+id/flag_favorito"
21        android:layout_toStartOf="@+id/flag_favorito"
22        android:gravity="center_vertical"
23        android:lines="2"
24        android:text="Nome
25        android:textAppearance="?android:attr/textAppearanceMedium"
26        android:textColor="@color/primary_text" />
27
28    <TextView
29        android:id="@+id/tipo_estab_lista_busca"
30        android:layout_width="match_parent"
31        android:layout_height="wrap_content"
32        android:layout_alignParentLeft="true"
33        android:layout_alignParentStart="true"
34        android:layout_below="@+id/nome_fantasia_lista_busca"
35        android:layout_marginBottom="5dp"
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\item_lista_estabelecimentos.xml

```
36     android:layout_toLeftOf="@+id/flag_fone"
37     android:layout_toStartOf="@+id/flag_fone"
38     android:gravity="center_vertical"
39     android:lines="2"
40     android:text="Tipo de estabelecimento"
41     android:textAppearance="?android:attr/textAppearanceMedium"
42     android:textColor="@color/secondary_text" />
43
44 <TextView
45     android:id="@+id/bairro_lista_busca"
46     android:layout_width="wrap_content"
47     android:layout_height="wrap_content"
48     android:layout_alignEnd="@+id/tipo_estab_lista_busca"
49     android:layout_alignParentLeft="true"
50     android:layout_alignParentStart="true"
51     android:layout_alignRight="@+id/tipo_estab_lista_busca"
52     android:layout_below="@+id/tipo_estab_lista_busca"
53     android:layout_marginBottom="5dp"
54     android:text="Bairro"
55     android:textAppearance="?android:attr/textAppearanceSmall"
56     android:textColor="@color/secondary_text" />
57
58 <TextView
59     android:id="@+id/municipio_lista_busca"
60     android:layout_width="wrap_content"
61     android:layout_height="wrap_content"
62     android:layout_alignParentLeft="true"
63     android:layout_alignParentStart="true"
64     android:layout_below="@+id/bairro_lista_busca"
65     android:text="Cidade"
66     android:textAppearance="?android:attr/textAppearanceSmall"
67     android:textColor="@color/secondary_text" />
68
69 <TextView
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\item_lista_estabelecimentos.xml

```
71     android:id="@+id/distancia_lista_busca"
72     android:layout_width="wrap_content"
73     android:layout_height="wrap_content"
74     android:layout_alignBaseline="@+id/municipio_lista_busca"
75     android:layout_alignBottom="@+id/municipio_lista_busca"
76     android:layout_toLeftOf="@+id/flag_gps"
77     android:layout_toStartOf="@+id/flag_gps"
78     android:textAppearance="?android:attr/textAppearanceSmall" />
79
80 <ImageButton
81     android:id="@+id/flag_favorito"
82     android:layout_width="wrap_content"
83     android:layout_height="wrap_content"
84     android:layout_alignParentEnd="true"
85     android:layout_alignParentRight="true"
86     android:layout_alignTop="@+id/nome_fantasia_lista_busca"
87     android:layout_centerHorizontal="true"
88     android:background="@android:color/transparent"
89     android:padding="16dp"
90     android:src="@drawable/ic_star_outline_grey600_24dp"
91     android:visibility="visible" />
92
93 <ImageButton
94     android:id="@+id/flag_fone"
95     android:layout_width="wrap_content"
96     android:layout_height="wrap_content"
97     android:layout_alignLeft="@+id/flag_favorito"
98     android:layout_alignStart="@+id/flag_favorito"
99     android:layout_centerVertical="true"
100    android:background="@android:color/transparent"
101    android:padding="16dp"
102    android:src="@drawable/ic_phone_grey600_24dp"
103    android:visibility="visible" />
104 <ImageButton
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\item_lista_estabelecimentos.xml

```
106     android:id="@+id/flag_gps"
107     android:layout_width="wrap_content"
108     android:layout_height="wrap_content"
109     android:layout_alignLeft="@+id/flag_fone"
110     android:layout_alignStart="@+id/flag_fone"
111     android:layout_below="@+id/tipo_estab_lista_busca"
112     android:background="@android:color/transparent"
113     android:padding="16dp"
114     android:src="@drawable/ic_crosshairs_gps_grey600_24dp"
115     android:visibility="visible" />
116
117 </RelativeLayout>
```


File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\item_listas.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     android:gravity="center_vertical"
6     android:orientation="vertical"
7     android:padding="16dp">
8
9     <TextView
10         android:id="@+id/item_linha_1"
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:clickable="false"
14         android:text="Medium Text"
15         android:textAppearance="?android:attr/textAppearanceMedium" />
16
17     <TextView
18         android:id="@+id/item_linha_2"
19         android:layout_width="match_parent"
20         android:layout_height="wrap_content"
21         android:clickable="false"
22         android:text="Small Text"
23         android:textAppearance="?android:attr/textAppearanceSmall"
24         android:textStyle="italic" />
25 </LinearLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\item_spinner.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2
3 <TextView xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@+id/item_spinner_busca_geral"
5     android:layout_width="match_parent"
6     android:layout_height="wrap_content"
7     android:gravity="center_vertical"
8     android:paddingBottom="16dp"
9     android:paddingLeft="8dp"
10    android:paddingRight="8dp"
11    android:paddingTop="16dp"
12    android:textAppearance="?android:attr/textAppearanceSmall"
13    android:textColor="@color/primary_text" />
14
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\preference_info.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:layout_margin="16dp"
6     android:background="@color/primary_light"
7     android:gravity="center"
8     android:orientation="vertical">
9
10    <ImageView
11        android:id="@+id/icone_pref_info"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:layout_gravity="center_horizontal"
15        android:layout_marginBottom="16dp"
16        android:layout_marginTop="32dp"
17        android:src="@drawable/ic_alert_grey600_48dp" />
18
19    <TextView
20        android:id="@+id/msg_pref_info"
21        android:layout_width="wrap_content"
22        android:layout_height="wrap_content"
23        android:layout_gravity="center_horizontal"
24        android:layout_marginBottom="32dp"
25        android:layout_marginLeft="16dp"
26        android:layout_marginRight="16dp"
27        android:text="@string/msg_info_preferencias"
28        android:textAlignment="center"
29        android:textAppearance="@android:style/TextAppearance.Small" />
30 </LinearLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\preference_seekbar.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:gravity="center_vertical"
6     android:orientation="vertical">
7
8     <TextView
9         style="?android:attr/textAppearanceMedium"
10        android:layout_width="match_parent"
11        android:layout_height="wrap_content"
12        android:layout_marginLeft="8dp"
13        android:layout_marginTop="8dp"
14        android:text="@string/distancia_estabelecimentos_titulo"
15        android:textColor="?android:attr/textColorSecondary" />
16
17    <TextView
18        style="?android:attr/textAppearanceSmall"
19        android:layout_width="match_parent"
20        android:layout_height="wrap_content"
21        android:layout_marginBottom="8dp"
22        android:layout_marginLeft="8dp"
23        android:text="@string/distancia_title_seekbar"
24        android:textColor="?android:attr/textColorSecondary" />
25
26    <LinearLayout
27        android:layout_width="match_parent"
28        android:layout_height="wrap_content"
29        android:layout_marginBottom="16dp">
30
31        <SeekBar
32            android:id="@+id/seekbar"
33            android:layout_width="0dp"
34            android:layout_height="wrap_content"
35            android:layout_weight="85" />
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\preference_seekbar.xml

```
36
37     <TextView
38         android:id="@android:id/summary"
39         android:layout_width="0dp"
40         android:layout_height="wrap_content"
41
42         android:layout_weight="15" />
43     </LinearLayout>
44 </LinearLayout>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\layout\view_dialog_busca_geral.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical"
6     android:padding="16dp">
7
8     <TextView
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:layout_margin="16dp"
12        android:text="@string/label_tipo_busca" />
13
14    <Spinner
15        android:id="@+id/combobox_tipo_busca_geral_dialog"
16        android:layout_width="match_parent"
17        android:layout_height="wrap_content"
18        android:layout_margin="16dp"
19        android:spinnerMode="dialog" />
20
21    <Spinner
22        android:id="@+id/combobox_busca_geral_dialog"
23        android:layout_width="match_parent"
24        android:layout_height="wrap_content"
25        android:layout_margin="16dp"
26        android:spinnerMode="dialog"
27        android:visibility="gone" />
28
29    <EditText
30        android:id="@+id/campo_busca_geral_dialog"
31        android:layout_width="match_parent"
32        android:layout_height="wrap_content"
33        android:layout_margin="16dp"
34        android:hint="Digite o termo"
35        android:visibility="gone" />
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\layout\view_dialog_busca_geral.xml

36

37 </LinearLayout>

File - E:\Android\Projetos\CadEsaude\app\src\main\res\menu\menu_consulta_gps.xml

```
1 <menu xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:app="http://schemas.android.com/apk/res-auto"
3     xmlns:tools="http://schemas.android.com/tools"
4     tools:context="br.com.jucemar_dimon.cadesaude.EstabelecimentoActivity">
5     <item
6         android:id="@+id/action_search"
7         android:icon="@drawable/ic_magnify_white_24dp"
8         android:iconifiedByDefault="true"
9         android:orderInCategory="100"
10        android:title="search"
11        android:visible="false"
12        app:actionViewClass="android.support.v7.widget.SearchView"
13        app:showAsAction="always|collapseActionView" />
14
15    <item
16        android:id="@+id/action_gps"
17        android:icon="@drawable/ic_crosshairs_gps_white_24dp"
18        android:orderInCategory="100"
19        android:title="GPS"
20        android:visible="false"
21        app:showAsAction="always" />
22 </menu>
23
```


File - E:\Android\Projetos\CadEsaude\app\src\main\res\menu\menu_main.xml

```
1 <menu xmlns:android="http://schemas.android.com/apk/res/android"
2
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     tools:context="br.com.jucemar_dimon.cadesaude.MainActivity">
6
7     <item
8         android:id="@+id/action_configuracoes"
9         android:orderInCategory="100"
10        android:title="@string/action_configuracoes"
11        app:showAsAction="never" />
12
13    <item
14        android:id="@+id/action_ajuda"
15        android:orderInCategory="100"
16        android:title="@string/action_Ajuda"
17        app:showAsAction="never" />
18
19    <item
20        android:id="@+id/action_sobre"
21        android:orderInCategory="100"
22        android:title="@string/action_sobre"
23        app:showAsAction="never" />
24
25    <item
26        android:id="@+id/action_search"
27        android:icon="@drawable/ic_magnify_white_24dp"
28        android:iconifiedByDefault="true"
29        android:orderInCategory="100"
30        android:title="search"
31        app:actionViewClass="android.support.v7.widget.SearchView"
32        app:showAsAction="always|collapseActionView" />
33    </menu>
34    .
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\menu\menu_primeira_configuracao.xml

```
1 <menu xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:app="http://schemas.android.com/apk/res-auto"
3     xmlns:tools="http://schemas.android.com/tools"
4     tools:context="br.com.jucemar_dimon.cadesaude.EstabelecimentoActivity">
5
6     <item
7         android:id="@+id/action_salvar_config"
8         android:icon="@drawable/ic_check_white_24dp"
9         android:orderInCategory="100"
10        android:title="Salvar"
11        app:showAsAction="always" />
12 </menu>
13
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\colors.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <color name="primary">#4CAF50</color>
4     <color name="primary_dark">#388E3C</color>
5     <color name="primary_light">#C8E6C9</color>
6     <color name="accent">#8bc34a</color>
7     <color name="primary_text">#212121</color>
8     <color name="secondary_text">#727272</color>
9     <color name="icons">#FFFFFF</color>
10    <color name="divider">#B6B6B6</color>
11    <color name="black_overlay">#660000</color>
12    <color name="fab_normal">#E53935</color>
13    <color name="fab_pressed">#FF5252</color>
14 </resources>
15
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
1 <resources>
2   <string name="app_name">CadESaúde</string>
3
4   <!-- Valores para as opções de menu principal-->
5   <string name="action_configuracoes">Configurações</string>
6   <string name="action_sobre">Sobre</string>
7   <string name="action_Ajuda">Ajuda</string>
8   <string name="titulo_config">Configurações</string>
9   <string name="titulo_ajuda">Ajuda</string>
10  <string name="titulo_sobre">Sobre</string>
11
12  <!-- Valores para as preferências do aplicativo-->
13  <string name="pref_municipios_preferidos_key">municipios_preferidos</string>
14  <string name="pref_distancia_estabelecimentos_key">distancia_estabelecimentos</string>
15  <string name="pref_info_key">info</string>
16  <string name="pref_estabelecimentos_favoritos_key">estabelecimentos_favoritos</string>
17  <string name="pref_estabelecimentos_favoritos_file">favoritos</string>
18  <string name="municipios_preferidos_titulo">Municípios
19  <string name="distancia_estabelecimentos_titulo">Distância
20  <string name="municipios_preferidos_subtitulo">Não há municípios selecionados</string>
21  <string name="municipios_dialog_title">selecione os municípios</string>
22  <string name="distancia_title_seekbar">Selecione a distância máxima</string>
23  <string-array name="municipios_preferidos_default" />
24
25  <string-array name="codigos_municipios">
26    <item>420005</item>
27    <item>420010</item>
28    <item>420020</item>
29    <item>420030</item>
30    <item>420040</item>
31    <item>420050</item>
32    <item>420055</item>
33    <item>420060</item>
34    <item>420070</item>
35  </string-array>
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
36 <item>420075</item>
37 <item>420080</item>
38 <item>420090</item>
39 <item>420100</item>
40 <item>420110</item>
41 <item>420120</item>
42 <item>420125</item>
43 <item>420127</item>
44 <item>420130</item>
45 <item>420140</item>
46 <item>420150</item>
47 <item>420160</item>
48 <item>420165</item>
49 <item>420170</item>
50 <item>420180</item>
51 <item>420190</item>
52 <item>420195</item>
53 <item>420205</item>
54 <item>420200</item>
55 <item>420207</item>
56 <item>421280</item>
57 <item>422000</item>
58 <item>420208</item>
59 <item>420209</item>
60 <item>420210</item>
61 <item>420213</item>
62 <item>420215</item>
63 <item>420220</item>
64 <item>420230</item>
65 <item>420240</item>
66 <item>420243</item>
67 <item>420250</item>
68 <item>420253</item>
69 <item>420257</item>
```

```
71 <item>420245</item>
72 <item>420270</item>
73 <item>420280</item>
74 <item>420285</item>
75 <item>420287</item>
76 <item>420290</item>
77 <item>420300</item>
78 <item>420310</item>
79 <item>420315</item>
80 <item>420320</item>
81 <item>420330</item>
82 <item>420340</item>
83 <item>420350</item>
84 <item>420360</item>
85 <item>420370</item>
86 <item>420380</item>
87 <item>420325</item>
88 <item>420390</item>
89 <item>420395</item>
90 <item>420400</item>
91 <item>420410</item>
92 <item>420415</item>
93 <item>420417</item>
94 <item>420419</item>
95 <item>420420</item>
96 <item>420425</item>
97 <item>420430</item>
98 <item>420435</item>
99 <item>420440</item>
100 <item>420445</item>
101 <item>420455</item>
102 <item>420450</item>
103 <item>420460</item>
104 <item>420470</item>
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
106 <item>420480</item>
107 <item>420490</item>
108 <item>420500</item>
109 <item>420510</item>
110 <item>420515</item>
111 <item>420517</item>
112 <item>420519</item>
113 <item>420520</item>
114 <item>420530</item>
115 <item>420535</item>
116 <item>420540</item>
117 <item>420543</item>
118 <item>420545</item>
119 <item>420550</item>
120 <item>420555</item>
121 <item>420560</item>
122 <item>420570</item>
123 <item>420580</item>
124 <item>420590</item>
125 <item>420600</item>
126 <item>420610</item>
127 <item>420620</item>
128 <item>420630</item>
129 <item>420640</item>
130 <item>420650</item>
131 <item>420660</item>
132 <item>420665</item>
133 <item>420670</item>
134 <item>420675</item>
135 <item>420680</item>
136 <item>420690</item>
137 <item>420700</item>
138 <item>420710</item>
139 <item>420720</item>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
141 <item>420740</item>
142 <item>420750</item>
143 <item>420757</item>
144 <item>420760</item>
145 <item>420765</item>
146 <item>420768</item>
147 <item>420770</item>
148 <item>420775</item>
149 <item>420780</item>
150 <item>420785</item>
151 <item>420790</item>
152 <item>420800</item>
153 <item>420810</item>
154 <item>420820</item>
155 <item>420830</item>
156 <item>420840</item>
157 <item>420845</item>
158 <item>420850</item>
159 <item>420860</item>
160 <item>420870</item>
161 <item>420880</item>
162 <item>420890</item>
163 <item>420895</item>
164 <item>420900</item>
165 <item>420910</item>
166 <item>420915</item>
167 <item>420917</item>
168 <item>420920</item>
169 <item>420930</item>
170 <item>420940</item>
171 <item>420945</item>
172 <item>420950</item>
173 <item>420960</item>
174 <item>420970</item>
```


File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
176     <item>420985</item>
177     <item>420990</item>
178     <item>421000</item>
179     <item>421003</item>
180     <item>421005</item>
181     <item>421010</item>
182     <item>421020</item>
183     <item>421030</item>
184     <item>421040</item>
185     <item>421050</item>
186     <item>421055</item>
187     <item>421060</item>
188     <item>421070</item>
189     <item>421080</item>
190     <item>421085</item>
191     <item>421090</item>
192     <item>421100</item>
193     <item>421105</item>
194     <item>421110</item>
195     <item>421120</item>
196     <item>421125</item>
197     <item>421130</item>
198     <item>421140</item>
199     <item>421145</item>
200     <item>421150</item>
201     <item>421160</item>
202     <item>421165</item>
203     <item>421170</item>
204     <item>421175</item>
205     <item>421180</item>
206     <item>421185</item>
207     <item>421187</item>
208     <item>421189</item>
209     <item>421190</item>
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
211 <item>421205</item>
212 <item>421210</item>
213 <item>421220</item>
214 <item>421223</item>
215 <item>421225</item>
216 <item>421227</item>
217 <item>421230</item>
218 <item>421240</item>
219 <item>421250</item>
220 <item>421260</item>
221 <item>421265</item>
222 <item>421270</item>
223 <item>421290</item>
224 <item>421300</item>
225 <item>421310</item>
226 <item>421315</item>
227 <item>421320</item>
228 <item>421330</item>
229 <item>421335</item>
230 <item>421340</item>
231 <item>421350</item>
232 <item>421360</item>
233 <item>421370</item>
234 <item>421380</item>
235 <item>421390</item>
236 <item>421400</item>
237 <item>421410</item>
238 <item>421415</item>
239 <item>421420</item>
240 <item>421430</item>
241 <item>421440</item>
242 <item>421450</item>
243 <item>421460</item>
244 <item>421480</item>
```

```
246 <item>421490</item>
247 <item>421500</item>
248 <item>421505</item>
249 <item>421507</item>
250 <item>421510</item>
251 <item>421520</item>
252 <item>421530</item>
253 <item>421535</item>
254 <item>421540</item>
255 <item>421545</item>
256 <item>421550</item>
257 <item>421555</item>
258 <item>421560</item>
259 <item>421565</item>
260 <item>421567</item>
261 <item>421568</item>
262 <item>421569</item>
263 <item>421570</item>
264 <item>421580</item>
265 <item>421575</item>
266 <item>421590</item>
267 <item>421600</item>
268 <item>421605</item>
269 <item>421610</item>
270 <item>421620</item>
271 <item>421630</item>
272 <item>421635</item>
273 <item>421625</item>
274 <item>421640</item>
275 <item>421650</item>
276 <item>421660</item>
277 <item>421670</item>
278 <item>421680</item>
279 <item>421690</item>
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
281 <item>421710</item>
282 <item>421715</item>
283 <item>421720</item>
284 <item>421725</item>
285 <item>421730</item>
286 <item>421740</item>
287 <item>421750</item>
288 <item>421755</item>
289 <item>421760</item>
290 <item>421770</item>
291 <item>421775</item>
292 <item>421780</item>
293 <item>421790</item>
294 <item>421795</item>
295 <item>421800</item>
296 <item>421810</item>
297 <item>421820</item>
298 <item>421825</item>
299 <item>421830</item>
300 <item>421835</item>
301 <item>421840</item>
302 <item>421850</item>
303 <item>421860</item>
304 <item>421870</item>
305 <item>421875</item>
306 <item>421880</item>
307 <item>421885</item>
308 <item>421890</item>
309 <item>421895</item>
310 <item>421900</item>
311 <item>421910</item>
312 <item>421915</item>
313 <item>421917</item>
314 <item>421920</item>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
316     <item>421935</item>
317     <item>421940</item>
318     <item>421950</item>
319     <item>421960</item>
320     <item>421970</item>
321     <item>421985</item>
322     </string-array>
323
324     <string-array name="nomes_municipios_view">
325         <item>Abdon Batista</item>
326         <item>Abelardo Luz</item>
327         <item>Agrolândia</item>
328         <item>Agronômica</item>
329         <item>Água Doce</item>
330         <item>Águas de Chapecó</item>
331         <item>Águas Frias</item>
332         <item>Águas Mornas</item>
333         <item>Alfredo Wagner</item>
334         <item>Alto Bela Vista</item>
335         <item>Anchieta</item>
336         <item>Angelina</item>
337         <item>Anita Garibaldi</item>
338         <item>Anitópolis</item>
339         <item>Antônio Carlos</item>
340         <item>Apiúna</item>
341         <item>Arabitã</item>
342         <item>Araquari</item>
343         <item>Araranguá</item>
344         <item>Armazém</item>
345         <item>Arroio Trinta</item>
346         <item>Arvoredo</item>
347         <item>Ascurra</item>
348         <item>Atalanta</item>
349         <item>Aurora</item>
350         <item>Balneário Arroio do Silva</item>
```

```
351 <item>Balneário Barra do Sul</item>
352 <item>Balneário Camboriú</item>
353 <item>Balneário Gaivota</item>
354 <item>Balneário Piçarras</item>
355 <item>Balneário Rincão</item>
356 <item>Bandeirante</item>
357 <item>Barra Bonita</item>
358 <item>Barra Velha</item>
359 <item>Bela Vista do Toldo</item>
360 <item>Belmonte</item>
361 <item>Benedito Novo</item>
362 <item>Biguaçu</item>
363 <item>Blumenau</item>
364 <item>Bocaina do Sul</item>
365 <item>Bom Jardim da Serra</item>
366 <item>Bom Jesus</item>
367 <item>Bom Jesus do Oeste</item>
368 <item>Bom Retiro</item>
369 <item>Bombinhas</item>
370 <item>Botuverá</item>
371 <item>Braço do Norte</item>
372 <item>Braço do Trombudo</item>
373 <item>Brunópolis</item>
374 <item>Brusque</item>
375 <item>Caçador</item>
376 <item>Caibi</item>
377 <item>Calmon</item>
378 <item>Camboriú</item>
379 <item>Campo Alegre</item>
380 <item>Campo Belo do Sul</item>
381 <item>Campo Erê</item>
382 <item>Campos Novos</item>
383 <item>Canelinha</item>
384 <item>Canoinhas</item>
---
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
386 <item>Capinzal</item>
387 <item>Capivari de Baixo</item>
388 <item>Catanduvas</item>
389 <item>Caxambu do Sul</item>
390 <item>Celso Ramos</item>
391 <item>Cerro Negro</item>
392 <item>Chapadão do Lageado</item>
393 <item>Chapecó</item>
394 <item>Cocal do Sul</item>
395 <item>Concórdia</item>
396 <item>Cordilheira Alta</item>
397 <item>Coronel Freitas</item>
398 <item>Coronel Martins</item>
399 <item>Correia Pinto</item>
400 <item>Corupá</item>
401 <item>Criciúma</item>
402 <item>Cunha Porã</item>
403 <item>Cunhataí</item>
404 <item>Curitibanos</item>
405 <item>Descanso</item>
406 <item>Dionísio Cerqueira</item>
407 <item>Dona Emma</item>
408 <item>Doutor Pedrinho</item>
409 <item>Entre Rios</item>
410 <item>Ermo</item>
411 <item>Erval Velho</item>
412 <item>Faxinal dos Guedes</item>
413 <item>Flor do Sertão</item>
414 <item>Florianópolis</item>
415 <item>Formosa do Sul</item>
416 <item>Forquilha</item>
417 <item>Fraiburgo</item>
418 <item>Frei Rogério</item>
419 <item>Galvão</item>
---
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
421 <item>Garuva</item>
422 <item>Gaspar</item>
423 <item>Governador Celso Ramos</item>
424 <item>Grão Pará</item>
425 <item>Gravatal</item>
426 <item>Guabiruba</item>
427 <item>Guaraciaba</item>
428 <item>Guaramirim</item>
429 <item>Guarujá do Sul</item>
430 <item>Guatambú</item>
431 <item>Herval d'Oeste</item>
432 <item>Ibiam</item>
433 <item>Ibicaré</item>
434 <item>Ibirama</item>
435 <item>Içara</item>
436 <item>Ilhota</item>
437 <item>Imarú</item>
438 <item>Imbituba</item>
439 <item>Imbuia</item>
440 <item>Indaial</item>
441 <item>Iomerê</item>
442 <item>Ipira</item>
443 <item>Iporã do Oeste</item>
444 <item>Ipuacu</item>
445 <item>Ipumirim</item>
446 <item>Iraceminha</item>
447 <item>Irani</item>
448 <item>Irati</item>
449 <item>Irinópolis</item>
450 <item>Itá</item>
451 <item>Itaiópolis</item>
452 <item>Itajaí</item>
453 <item>Itapema</item>
454 <item>Itapiranga</item>
```


File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
456 <item>Ituporanga</item>
457 <item>Jaborá</item>
458 <item>Jacinto Machado</item>
459 <item>Jaguaruna</item>
460 <item>Jaraguá do Sul</item>
461 <item>Jardinópolis</item>
462 <item>Joaçaba</item>
463 <item>Joinville</item>
464 <item>José Boiteux</item>
465 <item>Jupia</item>
466 <item>Lacerdópolis</item>
467 <item>Lages</item>
468 <item>Laguna</item>
469 <item>Lajeado Grande</item>
470 <item>Laurentino</item>
471 <item>Lauro Muller</item>
472 <item>Lebon Régis</item>
473 <item>Leoberto Leal</item>
474 <item>Lindóia do Sul</item>
475 <item>Lontras</item>
476 <item>Luiz Alves</item>
477 <item>Luzerna</item>
478 <item>Macieira</item>
479 <item>Mafra</item>
480 <item>Major Gercino</item>
481 <item>Major Vieira</item>
482 <item>Maracajá</item>
483 <item>Maravilha</item>
484 <item>Marema</item>
485 <item>Massaranduba</item>
486 <item>Matos Costa</item>
487 <item>Meleiro</item>
488 <item>Mirim Doce</item>
489 <item>Modelo</item>
```

```
491 <item>Monte Carlo</item>
492 <item>Monte Castelo</item>
493 <item>Morro da Fumaça</item>
494 <item>Morro Grande</item>
495 <item>Navegantes</item>
496 <item>Nova Erechim</item>
497 <item>Nova Itaberaba</item>
498 <item>Nova Trento</item>
499 <item>Nova Veneza</item>
500 <item>Novo Horizonte</item>
501 <item>Orleans</item>
502 <item>Otacílio Costa</item>
503 <item>Ouro</item>
504 <item>Ouro Verde</item>
505 <item>Paial</item>
506 <item>Painel</item>
507 <item>Palhoça</item>
508 <item>Palma Sola</item>
509 <item>Palmeira</item>
510 <item>Palmitos</item>
511 <item>Papanduva</item>
512 <item>Paraíso</item>
513 <item>Passo de Torres</item>
514 <item>Passos Maia</item>
515 <item>Paulo Lopes</item>
516 <item>Pedras Grandes</item>
517 <item>Penha</item>
518 <item>Peritiba</item>
519 <item>Pescaria Brava</item>
520 <item>Petrópolis</item>
521 <item>Pinhalzinho</item>
522 <item>Pinheiro Preto</item>
523 <item>Piratuba</item>
524 <item>Planalto Alegre</item>
---
```

```

526 <item>Ponte Alta</item>
527 <item>Ponte Alta do Norte</item>
528 <item>Ponte Serra</item>
529 <item>Porto Belo</item>
530 <item>Porto União</item>
531 <item>Pouso Redondo</item>
532 <item>Praia Grande</item>
533 <item>Presidente Castello Branco</item>
534 <item>Presidente Getúlio</item>
535 <item>Presidente Nereu</item>
536 <item>Princesa</item>
537 <item>Quilombo</item>
538 <item>Rancho Queimado</item>
539 <item>Rio das Antas</item>
540 <item>Rio do Campo</item>
541 <item>Rio do Oeste</item>
542 <item>Rio do Sul</item>
543 <item>Rio dos Cedros</item>
544 <item>Rio Fortuna</item>
545 <item>Rio Negrinho</item>
546 <item>Rio Rufino</item>
547 <item>Riqueza</item>
548 <item>Rodeio</item>
549 <item>Rommelândia</item>
550 <item>Salette</item>
551 <item>Saltinho</item>
552 <item>Salto Veloso</item>
553 <item>Sangão</item>
554 <item>Santa Cecília</item>
555 <item>Santa Helena</item>
556 <item>Santa Rosa de Lima</item>
557 <item>Santa Rosa do Sul</item>
558 <item>Santa Terezinha</item>
559 <item>Santa Terezinha do Progresso</item>
-- <item>Santiago do Sul</item>

```

```
561 <item>Santo Amaro da Imperatriz</item>
562 <item>São Bento do Sul</item>
563 <item>São Bernardino</item>
564 <item>São Bonifácio</item>
565 <item>São Carlos</item>
566 <item>São Cristóvão do Sul</item>
567 <item>São Domingos</item>
568 <item>São Francisco do Sul</item>
569 <item>São João Batista</item>
570 <item>São João do Itaperiú</item>
571 <item>São João do Oeste</item>
572 <item>São João do Sul</item>
573 <item>São Joaquim</item>
574 <item>São José</item>
575 <item>São José do Cedro</item> José
576 <item>São José do Cerrito</item>
577 <item>São Lourenço do Oeste</item>
578 <item>São Ludgero</item>
579 <item>São Martinho</item>
580 <item>São Miguel da Boa Vista</item>
581 <item>São Miguel do Oeste</item>
582 <item>São Pedro de Alcântara</item>
583 <item>Saudades</item>
584 <item>Schroeder</item>
585 <item>Seara</item>
586 <item>Serra Alta</item>
587 <item>Siderópolis</item>
588 <item>Sombrio</item>
589 <item>Sul Brasil</item>
590 <item>Taió</item>
591 <item>Tangará</item>
592 <item>Tigrinhos</item>
593 <item>Tijucas</item>
594 <item>Timbé do Sul</item>
595 <item>Timbó</item>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
596 <item>Timbó Grande</item>
597 <item>Três Barras</item>
598 <item>Treviço</item>
599 <item>Treze de Maio</item>
600 <item>Treze Tílias</item>
601 <item>Trombudo Central</item>
602 <item>Tubarão</item>
603 <item>Tunápolis</item>
604 <item>Turvo</item>
605 <item>União do Oeste</item>
606 <item>Urubici</item>
607 <item>Urupema</item>
608 <item>Urussanga</item>
609 <item>Vargão</item>
610 <item>Vargem</item>
611 <item>Vargem Bonita</item>
612 <item>Vidal Ramos</item>
613 <item>Videira</item>
614 <item>Vitor Meireles</item>
615 <item>Witmarsum</item>
616 <item>Xanxerê</item>
617 <item>Xavantina</item>
618 <item>Xaxim</item>
619 <item>Zortéa</item>
620 </string-array>
621
622 <string-array name="nomes_municipios_bd">
623 <item>ABDON BATISTA</item>
624 <item>ABELARDO LUZ</item>
625 <item>AGROLÂNDIA</item>
626 <item>AGRONOMICA</item>
627 <item>AGUA DOCE</item>
628 <item>AGUAS DE CHAPECO</item>
629 <item>AGUAS FRIAS</item>
630 <item>AGUAS MORNAS</item>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
631 <item>ALFREDO WAGNER</item>
632 <item>ALTO BELA VISTA</item>
633 <item>ANCHIETA</item>
634 <item>ANGELINA</item>
635 <item>ANITA GARIBALDI</item>
636 <item>ANITAPOLIS</item>
637 <item>ANTONIO CARLOS</item>
638 <item>APIUNA</item>
639 <item>ARABUTA</item>
640 <item>ARAQUARI</item>
641 <item>ARARANGUA</item>
642 <item>ARMAZEM</item>
643 <item>ARROIO TRINTA</item>
644 <item>ARVOREDO</item>
645 <item>ASCURRA</item>
646 <item>ATALANTA</item>
647 <item>ATIORA</item>
648 <item>BALNEARIO ARROIO DO SILVA</item>
649 <item>BALNEARIO BARRA DO SUL</item>
650 <item>BALNEARIO CAMBORIU</item>
651 <item>BALNEARIO GAIVOTA</item>
652 <item>BALNEARIO PICARRAS</item>
653 <item>BALNEARIO RINCAO</item>
654 <item>BANDEIRANTE</item>
655 <item>BARRA BONITA</item>
656 <item>BARRA VELHA</item>
657 <item>BELA VISTA DO TOLDOS</item>
658 <item>BELMONTE</item>
659 <item>BENEDITO NOVO</item>
660 <item>BIGUACU</item>
661 <item>BLUMENAU</item>
662 <item>BOCAINA DO SUL</item>
663 <item>BOM JARDIM DA SERRA</item>
664 <item>BOM JESUS</item>
-- <item>BOM JESUS DO OESTE</item>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
666 <item>BOM RETIRO</item>
667 <item>BOMBINHAS</item>
668 <item>BOTUVERA</item>
669 <item>BRACO DO NORTE</item>
670 <item>BRACO DO TROMBUDO</item>
671 <item>BRUNOPOLIS</item>
672 <item>BRUSQUE</item>
673 <item>CACADOR</item>
674 <item>CAIBI</item>
675 <item>CALMON</item>
676 <item>CAMBORIU</item>
677 <item>CAMPO ALEGRE</item>
678 <item>CAMPO BELO DO SUL</item>
679 <item>CAMPO ERE</item>
680 <item>CAMPOS NOVOS</item>
681 <item>CANELINHA</item>
682 <item>CANOINHAS</item>
683 <item>CAPAO ALTO</item>
684 <item>CAPINZAL</item>
685 <item>CAPIVARI DE BAIXO</item>
686 <item>CATANDUVAS</item>
687 <item>CAXAMBU DO SUL</item>
688 <item>CELSO RAMOS</item>
689 <item>CERRO NEGRO</item>
690 <item>CHAPADAO DO LAGEADO</item>
691 <item>CHAPECO</item>
692 <item>COCAL DO SUL</item>
693 <item>CONCORDIA</item>
694 <item>CORDILHEIRA ALTA</item>
695 <item>CORONEL FREITAS</item>
696 <item>CORONEL MARTINS</item>
697 <item>CORREIA PINTO</item>
698 <item>CORUPA</item>
699 <item>CRICIUMA</item>
---
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
701 <item>CUNHATAI</item>
702 <item>CURITIBANOS</item>
703 <item>DESCANSO</item>
704 <item>DIONISIO CERQUEIRA</item>
705 <item>DONA EMMA</item>
706 <item>DOUTOR PEDRINHO</item>
707 <item>ENTRE RIOS</item>
708 <item>ERMO</item>
709 <item>ERVAL VELHO</item>
710 <item>FAXINAL DOS GUEDES</item>
711 <item>FLOR DO SERTAO</item>
712 <item>FLORIANOPOLIS</item>
713 <item>FORMOSA DO SUL</item>
714 <item>FORQUILHINHA</item>
715 <item>FRAIBURGO</item>
716 <item>FREI ROGERIO</item>
717 <item>GALVAO</item>
718 <item>GAROPABA</item>
719 <item>GARUVA</item>
720 <item>GASPAR</item>
721 <item>GOVERNADOR CELSO RAMOS</item>
722 <item>GRAO PARA</item>
723 <item>GRAVATAL</item>
724 <item>GUABIRUBA</item>
725 <item>GUARACIABA</item>
726 <item>GUARAMIRIM</item>
727 <item>GUARUJA DO SUL</item>
728 <item>GUATAMBU</item>
729 <item>HERVAL D OESTE</item>
730 <item>IBIAM</item>
731 <item>IBICARE</item>
732 <item>IBIRAMA</item>
733 <item>ICARA</item>
734 <item>ILHOTA</item>
```



```
736 <item>IMBITUBA</item>
737 <item>IMBUIA</item>
738 <item>INDAIAL</item>
739 <item>IOMERE</item>
740 <item>IPIRA</item>
741 <item>IPORA DO OESTE</item>
742 <item>IPUACU</item>
743 <item>IPUMIRIM</item>
744 <item>IRACEMINHA</item>
745 <item>IRANI</item>
746 <item>IRATI</item>
747 <item>IRINEOPOLIS</item>
748 <item>ITA</item>
749 <item>ITAIOPOLIS</item>
750 <item>ITAJAI</item>
751 <item>ITAPEMA</item>
752 <item>ITAPIRANGA</item>
753 <item>ITAPOA</item>
754 <item>ITUPORANGA</item>
755 <item>JABORA</item>
756 <item>JACINTO MACHADO</item>
757 <item>JAGUARUNA</item>
758 <item>JARAGUA DO SUL</item>
759 <item>JARDINOPOLIS</item>
760 <item>JOACABA</item>
761 <item>JOINVILLE</item>
762 <item>JOSE BOITEUX</item>
763 <item>JUPIA</item>
764 <item>LACERDOPOLIS</item>
765 <item>LAGES</item>
766 <item>LAGUNA</item>
767 <item>LAJEADO GRANDE</item>
768 <item>LAURENTINO</item>
769 <item>LAURO MULLER</item>
```

```
771 <item>LEOBERTO LEAL</item>
772 <item>LINDOIA DO SUL</item>
773 <item>LONTRAS</item>
774 <item>LUIZ ALVES</item>
775 <item>LUZERNA</item>
776 <item>MACIEIRA</item>
777 <item>MAFRA</item>
778 <item>MAJOR GERCINO</item>
779 <item>MAJOR VIEIRA</item>
780 <item>MARACAJA</item>
781 <item>MARAVILHA</item>
782 <item>MAREMA</item>
783 <item>MASSARANDUBA</item>
784 <item>MATOS COSTA</item>
785 <item>MELEIRO</item>
786 <item>MIRIM DOCE</item>
787 <item>MODELO</item>
788 <item>MONDAI</item>
789 <item>MONTE CARLO</item>
790 <item>MONTE CASTELO</item>
791 <item>MORRO DA FUMACA</item>
792 <item>MORRO GRANDE</item>
793 <item>NAVEGANTES</item>
794 <item>NOVA ERECHIM</item>
795 <item>NOVA ITABERABA</item>
796 <item>NOVA TRENTO</item>
797 <item>NOVA VENEZA</item>
798 <item>NOVO HORIZONTE</item>
799 <item>ORLEANS</item>
800 <item>OTACILIO COSTA</item>
801 <item>OURO</item>
802 <item>OURO VERDE</item>
803 <item>PAIAL</item>
804 <item>PAINEL</item>
-- <item>PALHOCA</item>
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
806 <item>PALMA SOLA</item>
807 <item>PALMEIRA</item>
808 <item>PALMITOS</item>
809 <item>PAPANDUVA</item>
810 <item>PARAISO</item>
811 <item>PASSO DE TORRES</item>
812 <item>PASSOS MAIA</item>
813 <item>PAULO LOPES</item>
814 <item>PEDRAS GRANDES</item>
815 <item>PENHA</item>
816 <item>PERITIBA</item>
817 <item>PESCARIA BRAVA</item>
818 <item>PETROLANDIA</item>
819 <item>PINHALZINHO</item>
820 <item>PINHEIRO PRETO</item>
821 <item>PIRATUBA</item>
822 <item>PLANALTO ALEGRE</item>
823 <item>POMERODE</item>
824 <item>PONTE ALTA</item>
825 <item>PONTE ALTA DO NORTE</item>
826 <item>PONTE SERRADA</item>
827 <item>PORTO BELO</item>
828 <item>PORTO UNIAO</item>
829 <item>POUSO REDONDO</item>
830 <item>PRAIA GRANDE</item>
831 <item>PRESIDENTE CASTELLO BRANCO</item>
832 <item>PRESIDENTE GETULIO</item>
833 <item>PRESIDENTE NEREU</item>
834 <item>PRINCESA</item>
835 <item>QUILOMBO</item>
836 <item>RANCHO QUEIMADO</item>
837 <item>RIO DAS ANTAS</item>
838 <item>RIO DO CAMPO</item>
839 <item>RIO DO OESTE</item>
... <item>RIO DO SUL</item>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
841 <item>RIO DOS CEDROS</item>
842 <item>RIO FORTUNA</item>
843 <item>RIO NEGRINHO</item>
844 <item>RIO RUFINO</item>
845 <item>RIQUEZA</item>
846 <item>RODEIO</item>
847 <item>ROMELANDIA</item>
848 <item>SALETE</item>
849 <item>SALTINHO</item>
850 <item>SALTO VELOSO</item>
851 <item>SANGAO</item>
852 <item>SANTA CECILIA</item>
853 <item>SANTA HELENA</item>
854 <item>SANTA ROSA DE LIMA</item>
855 <item>SANTA ROSA DO SUL</item>
856 <item>SANTA TEREZINHA</item>
857 <item>SANTA TEREZINHA DO PROGRESSO</item>
858 <item>SANTIAGO DO SUL</item>
859 <item>SANTO AMARO DA IMPERATRIZ</item>
860 <item>SAO BENTO DO SUL</item>
861 <item>SAO BERNARDINO</item>
862 <item>SAO BONIFACIO</item>
863 <item>SAO CARLOS</item>
864 <item>SAO CRISTOVAO DO SUL</item>
865 <item>SAO DOMINGOS</item>
866 <item>SAO FRANCISCO DO SUL</item>
867 <item>SAO JOAO BATISTA</item>
868 <item>SAO JOAO DO ITAPERIU</item>
869 <item>SAO JOAO DO OESTE</item>
870 <item>SAO JOAO DO SUL</item>
871 <item>SAO JOAQUIM</item>
872 <item>SAO JOSE</item>
873 <item>SAO JOSE DO CEDRO</item> JOSE
874 <item>SAO DO CERRITO</item>
-- <item>SAO LOURENCO DO OESTE</item>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
876 <item>SAO LUDGERO</item>
877 <item>SAO MARTINHO</item>
878 <item>SAO MIGUEL DA BOA VISTA</item>
879 <item>SAO MIGUEL DO OESTE</item>
880 <item>SAO PEDRO DE ALCANTARA</item>
881 <item>SAUDADES</item>
882 <item>SCHROEDER</item>
883 <item>SEARA</item>
884 <item>SERRA ALTA</item>
885 <item>SIDEROPOLIS</item>
886 <item>SOMBRIO</item>
887 <item>SUL BRASIL</item>
888 <item>TAIO</item>
889 <item>TANGARA</item>
890 <item>TIGRINHOS</item>
891 <item>TIJUCAS</item>
892 <item>TIMBE DO SUL</item>
893 <item>TIMBO</item>
894 <item>TIMBO GRANDE</item>
895 <item>TRES BARRAS</item>
896 <item>TREVISO</item>
897 <item>TREZE DE MAIO</item>
898 <item>TREZE TILIAS</item>
899 <item>TROMBUDO CENTRAL</item>
900 <item>TUBARAO</item>
901 <item>TUNAPOLIS</item>
902 <item>TURVO</item>
903 <item>UNIAO DO OESTE</item>
904 <item>URUBICI</item>
905 <item>URUPEMA</item>
906 <item>URUSSANGA</item>
907 <item>VARGEAO</item>
908 <item>VARGEM</item>
909 <item>VARGEM BONITA</item>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
911 <item>VIDEIRA</item>
912 <item>VITOR MEIRELES</item>
913 <item>WITMARSUM</item>
914 <item>XANXERE</item>
915 <item>XAVANTINA</item>
916 <item>XAXIM</item>
917 <item>ZORTEA</item>
918 </string-array>
919
920 <string-array name="tipos_etabelecimento">
921 <item>CENTRAL DE NOTIFICACAO,CAPTACAO E DISTRIB DE ORGAOS ESTADUAL</item>
922 <item>CENTRO DE ATENCAO HEMOTERAPIA E OU HEMATOLOGICA</item>
923 <item>CENTRO DE APOIO A SAUDE DA FAMILIA</item>
924 <item>CENTRO DE ATENCAO PSICOSSOCIAL</item>
925 <item>CENTRO DE PARTO NORMAL - ISOLADO</item>
926 <item>CENTRO DE SAUDE/UNIDADE BASICA</item>
927 <item>CLINICA/CENTRO DE ESPECIALIDADE</item>
928 <item>CONSULTORIO ISOLADO</item>
929 <item>HOSPITAL ESPECIALIZADO</item>
930 <item>HOSPITAL GERAL</item>
931 <item>HOSPITAL/DIA - ISOLADO</item>
932 <item>LABORATORIO CENTRAL DE SAUDE PUBLICA LACEN</item>
933 <item>LABORATORIO DE SAUDE PUBLICA</item>
934 <item>POLICLINICA</item>
935 <item>POSTO DE SAUDE</item>
936 <item>PRONTO ATENDIMENTO</item>
937 <item>PRONTO SOCORRO ESPECIALIZADO</item>
938 <item>PRONTO SOCORRO GERAL</item>
939 <item>UNIDADE DE APOIO DIAGNOSE E TERAPIA (SADT ISOLADO)</item>
940 <item>UNIDADE MISTA</item>
941 </string-array>
942
943 <string-array name="tipos_atendimento">
944 <item>AMBULATORIAL</item>
945 <item>INTERNAÇÃO</item>
...

```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
946     <item>SADT</item>
947     <item>URGENCIA</item>
948
949 </string-array>
950
951 <string name="title_activity_estabelecimento">EstabelecimentoActivity</string>
952 <string name="action_settings">Settings</string>
953 <string name="txt_lista_sem_favoritos">Você não possui favoritos</string>
954
955 <!-- Strings de BuscaGpsActivity -->
956 <string name="fab_filtro_tipo_estabelecimento">Tipo de estabelecimento</string>
957 <string name="fab_filtro_tipo_atendimento">Tipo de atendimento</string>
958 <string name="fab_filtro_tipo_convênio">Tipo de convênio</string>
959 <string name="fab_filtro_tipo_servico">Tipo de serviço</string>
960 <string name="fab_filtro_tipo_profissional">Tipo de profissional</string>
961 <string name="fab_filtro_limpar_todos">Limpar filtros</string>
962 <string name="txt_erro_gps">Verifique seu GPS</string>
963 <string name="titulo_busca_gps_activity">Consulta por proximidade</string>
964
965 <!-- String FiltroGeralDialog -->
966 <string name="titulo_dialog_filtro">Selecione os itens a serem filtrados</string>
967
968 <!-- Strings de BuscaGpsActivity Snackbar-->
969 <string name="tentar_novamente_action">TENTAR NOVAMENTE</string>
970 <string name="tentar_novamente_label">Localização obrigatória</string>
971
972 <string-array name="tipos_convênio">
973     <item>PARTICULAR</item>
974     <item>PLANO / SEGURO PROPRIO</item>
975     <item>PLANO / SEGURO TERCEIRO</item>
976     <item>PLANO DE SAUDE PRIVADO</item>
977     <item>PLANO DE SAUDE PUBLICO</item>
978     <item>SUS</item>
979 </string-array>
---
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
981 <string name="title_activity_busca_geral">Busca geral</string>
982 <string name="txt_error_busca_geral">Nenhum resultado localizado</string>
983
984 <!-- BuscaGeralActivity -->
985 <string name="label_tipo_busca">Selecione um tipo de busca:</string>
986 <string name="titulo_dialog_buscar_geral">Busca geral</string>
987 <string-array name="tipos_buscas">
988     <item>Selecione uma opção</item>
989     <item>Tipos de convênios</item>
990     <item>Formas de atendimento</item>
991     <item>Nome do profissional</item>
992     <item>Especialidades</item>
993     <item>Nome do estabelecimento</item>
994     <item>Serviços oferecidos</item>
995 </string-array>
996
997 <!-- SobreActivity -->
998 <string name="appName">CadESaúde</string>
999
1000 <string name="appDesc">Aplicativo desenvolvido por Jucenar Dimon como trabalho de conclusão do curso de Sistemas
de Informações da Universidade Federal de Santa Catarina(UFSC). Este aplicativo tem como
objetivo não somente a aplicação dos conhecimentos adquiridos ao longo do curso como também a criação de uma
ferramenta para auxiliar o cidadão catarinense na busca por estabelecimentos de saúde na sua região.</string>
1001 <string name="appVersao">Versão do aplicativo</string>
1002 <string name="bdVersao">Versão do Banco de dados</string>
1003 <string name="title_activity_primeiros_passos">PrimeirosPassosActivity</string>
1004 <string name="dummy_button">Configurações</string>
1005 <string name="dummy_content">Seja bem vndo ao aplicativo CadESaúde.\n Neste seu primeiro acesso você deverá
clicar no botão configurações e selecionar os municípios que deseja incluir nas buscas por estabelecimentos de saúde.
</string>
1006
1007 <!-- AjudaActivity - Tela inicial -->
1008 <string name="cabecalho_ajuda">Neste menu do aplicativo você poderá tirar algumas dúvidas sobre termos
específicos que são utilizados no sistema de cadastros de estabelecimentos de saúde do DATASUS.</string>
1009 <string name="label_btn_tipos_estabelecimentos">Tipos de estabelecimentos</string>
1010 <string name="label_btn_tipos_servicos">Tipos de serviços existentes</string>
```


File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
1011 <!-- AjudaActivity - Tipos de estabelecimentos -->
1012 <string name="titulo_actvt_ajuda_estabelecimentos">Tipos de estabelecimentos</string>
1013
1014
1015 <string name="posto_de_saude_titulo">Posto de Saúde</string>
1016 <string name="posto_de_saude">Unidade destinada à prestação de assistência a uma determinada população, de forma
    programada ou não, por profissional de nível médio, com a presença intermitente ou não do profissional médico.</
string>
1017
1018 <string name="centro_saude_titulo">Centro de Saúde/Unidade Básica de Saúde</string>
1019 <string name="centro_saude">Unidade para realização de atendimentos de atenção básica e integral a uma população
    , de forma programada ou não, nas especialidades básicas, podendo oferecer assistência odontológica e de outros
    profissionais de nível superior. A assistência deve ser permanente e prestada por médico generalista ou especialista
    nestas áreas. Podendo ou não oferecer: SADT e Pronto atendimento 24 Horas.</string>
1020
1021 <string name="policlinica_titulo">Policlínica</string>
1022 <string name="policlinica">Unidade de saúde para prestação de atendimento ambulatorial em várias especialidades,
    incluindo ou não as especialidades básicas, podendo ainda ofertar outras especialidades não médicas. Podendo ou não
    oferecer: SADT e Pronto atendimento 24 Horas.</string>
1023
1024 <string name="hospital_geral_titulo">Hospital Geral</string>
1025 <string name="hospital_geral">Hospital destinado à prestação de atendimento nas especialidades básicas, por
    especialistas e/ou outras especialidades médicas. Pode dispor de serviço de Urgência/Emergência. Deve dispor também
    de SADT de média complexidade. Podendo Ter ou não SIPAC.</string>
1026
1027 <string name="hospital_especializado_titulo">Hospital Especializado</string>
1028 <string name="hospital_especializado">Hospital destinado à prestação de assistência à saúde em uma única
    especialidade/área. Pode dispor de serviço de Urgência/Emergência e SADT. Podendo Ter ou não SIPAC Geralmente de
    referência regional, macro regional ou estadual.</string>
1029
1030 <string name="unidade_mista_titulo">Unidade Mista</string>
1031 <string name="unidade_mista">Unidade de saúde básica destinada à prestação de atendimento em atenção básica e
    integral à saúde, de forma programada ou não, nas especialidades básicas, podendo oferecer assistência odontológica e
    de outros profissionais, com unidade de internação, sob administração única. A assistência médica deve ser
    permanente e prestada por médico especialista ou generalista. Pode dispor de urgência/emergência e SADT básico ou de
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
1031   rotina.</string>
1032
1033   <string name="pronto_socorro_geral_titulo">Pronto Socorro Geral</string>
1034   <string name="pronto_socorro_geral">Unidade destinada à prestação de assistência a pacientes com ou sem risco de
vida, cujos agravos necessitam de atendimento imediato. Podendo ter ou não internação.</string>
1035
1036   <string name="pronto_socorro_especializado_titulo">Pronto Socorro Geral</string>
1037   <string name="pronto_socorro_especializado">Unidade destinada à prestação de assistência em uma ou mais
especialidades, a pacientes com ou sem risco de vida, cujos agravos necessitam de atendimento imediato.</string>
1038
1039   <string name="consultorio_isolado_titulo">Consultório Isolado</string>
1040   <string name="consultorio_isolado">Sala isolada destinada à prestação de assistência médica ou odontológica ou
de outros profissionais de saúde de nível superior.</string>
1041
1042   <string name="clinica_titulo">Clínica/Centro de Especialidade</string>
1043   <string name="clinica">Clínica Especializada destinada à assistência ambulatorial em apenas uma especialidade/
área da assistência.</string>
1044
1045   <string name="sadt_titulo">Unidade de Serviço de Apoio de Diagnose e Terapia (SADT ISOLADO)</string>
1046   <string name="sadt">Unidades isoladas onde são realizadas atividades que auxiliam a determinação de diagnóstico
e/ou complementam o tratamento e a reabilitação do paciente.</string>
1047
1048   <string name="parto_isolado_titulo">Centro de Parto Normal Isolado</string>
1049   <string name="parto_isolado">Unidade intra-hospitalar ou isolada, especializada no atendimento da mulher no
período gravídico puerperal.</string>
1050
1051   <string name="hospital_dia_titulo">Hospital/Dia - Isolado</string>
1052   <string name="hospital_dia">Unidades especializadas no atendimento de curta duração com caráter intermediário
entre a assistência ambulatorial e a internação.</string>
1053
1054   <string name="labo_saude_publica_titulo">Laboratório Central de Saúde Pública</string>
1055   <string name="labo_saude_publica">Estabelecimento de Saúde que integra o Sistema Nacional de Laboratórios de
Saúde Pública - SISLAB, em conformidade com normalização vigente.</string>
1056
1057   <string name="centro_hemato_titulo">Centro de Atencao Hemoterapica e ou Hematologica</string>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
1058     <string name="centro_hemato">Estabelecimento que realiza o ciclo do sangue, desde a captação do doador,
1059     processamento, testes sorológicos, testes imunohematológicos, distribuição e transfusão de sangue de maneira total ou
1060     parcial.</string>
1061     <string name="caps_titulo">Centro De Atenção Psicossocial (CAPS)</string>
1062     <string name="caps">Entende-se por Centro de Atenção Psicossocial o tipo de estabelecimento isolado com as
1063     especificações seguintes: Estabelecimento de saúde de esfera administrativa pública com serviços de saúde mental
1064     abertos e comunitários do Sistema Único de Saúde SUS destinados a prestar atenção diária às pessoas que sofrem com
1065     transtornos mentais severos e persistentes, em sua área territorial, em regime de tratamento intensivo, semi-
1066     intensivo e não-intensivo. Devem oferecer cuidados clínicos e de reabilitação psicossocial e têm como objetivo
1067     substituir o modelo hospitalocêntrico, evitar internações, favorecer o exercício da cidadania e da inclusão social
1068     dos usuários e de suas famílias, e executar as atribuições de supervisão e regulamentação de rede de serviços de
1069     saúde mental, quando de natureza pública.</string>
1070     <string name="upa_titulo">Pronto Atendimento</string>
1071     <string name="upa">Estabelecimento autônomo não-hospitalar, que integra a Rede de Atenção às Urgências e
1072     Emergências, destinado à assistência aos pacientes acometidos por quadros de urgência e emergência, realizando o
1073     atendimento inicial, estabilizando o paciente e definindo a necessidade ou não de encaminhamento a serviços de maior
1074     complexidade.</string>
1075     <!-- AjudaActivity - Serviços especializados -->
1076     <string name="titulo_actvt_ajuda_servicos">Tipos de serviços</string>
1077     <string name="I01c1">Atenção à Doença Renal Crônica</string>
1078     <string name="I01c2">Tratamento Dialítico-Hemodialise; Confecção/Intervenção de Acessos para Diálise; Tratamento
1079     Nefrologia em Geral; Tratamento Dialítico Peritonial; Tratamento em DRC</string>
1080     <string name="I02c1">Atenção Às Pessoas com Doenças Raras</string>
1081     <string name="I02c2">Atenção Especializada Em Doenças Raras; Referência em Doenças Raras</string>
1082     <string name="I03c1">Atenção às Pessoas em Situação de Violência Sexual</string>
1083     <string name="I03c2">Referência para Atenção Integral às Pessoas em Situação de Violência Sexual; Atenção
1084     Integral à Saúde de Adolescentes em Situação de Violência Sexual; Atenção Integral à Saúde de Crianças em Situação de
1085     Violência Sexual; Atenção Integral à Saúde de Homens em Situação de Violência Sexual; Atenção Integral à Saúde de
1086     Pessoas Idosas em Situação de Violência Sexual; Interrupção de Gravidez nos Casos Previstos em Lei; Atenção
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
1076 Ambulatorial às Pessoas em Situação de Violência Sexual;Coleta de Vestígios de Violência Sexual</string>
1077
1078 <string name="104c1">Atenção Básica</string>
1079 <string name="104c2">Atenção Básica com Saúde Bucal</string>
1080
1081 <string name="105c1">Atenção Especializada no Processo Transsexualizador</string>
1082 <string name="105c2">Acompanhamento Clínico, pré e pós operatório e hormonioterapia; Cirurgia e Acompanhamento
    pré e pós oratório</string>
1083
1084 <string name="106c1">Atendimento Itinerante de Assistência e Ensino em Saúde</string>
1085 <string name="106c2">ODONTOLOGIA - PROJETO SAÚDE NA ESCOLA (PSE); ODONTOLOGIA - PROJETO BRASIL ALFABETIZADO (PBA
    ); OFTALMOLOGIA - PROJETO SAÚDE NA ESCOLA (PSE); OFTALMOLOGIA - PROJETO BRASIL ALFABETIZADO (PBA)</string>
1086
1087 <string name="107c1">Cirurgia Vascular</string>
1088 <string name="107c2">Fístula arteriovenosa sem enxerto; Fístula arteriovenosa com enxerto</string>
1089
1090 <string name="108c1">Diagnóstico por Imagem</string>
1091 <string name="108c2">Radiologia; Ultra-sonografia; Tomografia computadorizada; Ressonância magnética; Radiologia
    intervencionista; Radiologia por telemedicina; Ultra-sonografia por telemedicina; Tomografia computadorizada por
    telemedicina;Ressonância magnética por telemedicina; Radiologia intervencionista por telemedicina; Mamografia;
    Mamografia por Telemedicina</string>
1092
1093 <string name="109c1">Estratégia de Saúde da Família</string>
1094 <string name="109c2">Saúde da família; Saúde bucal MI; Saúde bucal MII</string>
1095
1096 <string name="110c1">Estratégias de Agentes Comunitário de Saúde</string>
1097 <string name="110c2">Estratégia de Agentes Comunitários de Saúde - EACS; Estratégia de Agentes Comunitários de
    Saúde com Saúde Bucal - (modalidade I); Estratégia de Agentes Comunitários de Saúde com Saúde Bucal - (modalidade II)
    </string>
1098
1099 <string name="111c1">Hospital-Dia</string>
1100 <string name="111c2">Saúde mental; AIDS; Geriátrico; Fibrose cística; Cirúrgico / Diagnóstico; Acompanhamento pós
    -transplante de medula óssea</string>
1101
1102 <string name="112c1">Medicina Nuclear</string>
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\values\strings.xml

```
1103     <string name="112c2">Medicina nuclear in vivo; Medicina nuclear in vitro; Medicina nuclear in vivo por
1104     telemedicina</string>
1105     <string name="113c1">Regulação Assistencial dos Serviços de Saúde</string>
1106     <string name="113c2">Central de Regulação de Internações Hospitalares; Cental de Regulação de Consultas e Exames;
1107     Central de Regulação de Urgências; Unidade de autorizações de TFD intermunicipais; Unidade de autorizações de TFD
1108     interestaduais; Central Estadual de Regulação da Alta Complexidade</string>
1109     <string name="114c1">Serviço de Apoio a Saúde da Família</string>
1110     <string name="114c2">NASF 1; NASF 2</string>
1111     <string name="115c1">Serviço de Atenção a Saúde do Trabalhador</string>
1112     <string name="115c2">Atendimento/acompanhamento em saúde do trabalhador; Centro de referência em saúde do
1113     trabalhador CEREST estadual; VIGILÂNCIA EM SAÚDE DO TRABALHADOR (VISAT)</string>
1114     <string name="116c1">Serviço de Atenção a DST/HIV/AIDS</string>
1115     <string name="116c2">Testagem e aconselhamento; Diagnóstico e tratamento; Diagnóstico, tratamento e capacitação
1116     técnica</string>
1117     <string name="117c1">Serviço de Atenção à Obesidade</string>
1118     <string name="117c2">Tratamento clínico, cirúrgico, reparador e acompanhamento ao obeso</string>
1119
1120     <string name="118c1">Serviço de Atenção a Saúde Auditiva</string>
1121     <string name="118c2">Serviço de Atenção à Saúde Auditiva na Média Complexidade; Serviço de Atenção à Saúde
1122     Auditiva na Alta Complexidade; Terapia fonoaudiológica; Diagnóstico em audiologia/otologia; Implante Coclear; Triagem
1123     Auditiva Neonatal; Diagnóstico em Audiologia/Otologia po Telemedicina; Atenção Especializada às Pessoas com
1124     Deficiência Auditiva</string>
1125
1126     <string name="119c1">Serviço de Atenção á Saúde das Pessoas Ostomizadas</string>
1127     <string name="119c2">Atenção as Pessoas Ostomizadas I; Atenção as Pessoas Ostomizadas II</string>
1128
1129     <string name="120c1">Serviço de Atenção a Saúde dos Adolescentes em Conflito com a Lei, em Regime de Internação e
1130     Internação Provisória</string>
1131     <string name="120c2">Unidade sócio-educativa com até 40 adolescentes; Unidade sócio-educativa com de 41 a 100
1132     adolescentes; Unidade sócio-educativa com de 101 a 180 adolescentes; Unidade sócio-educativa com mais de 180
```



```

1127 adoIcetes</string>
1128
1129 <string name="121cl">Serviço de Atenção a Saúde no Sistema Penitenciário</string>
1130 <string name="121c2">Atendimento em presídio até 100 presos; Atendimento em manicômio até 100 presos;
1131 Atendimento em presídio acima de 100 presos; Atendimento em manicômio acima de 100 presos</string>
1132
1132 <string name="122cl">Serviço de Atenção a Saúde Reprodutiva</string>
1133 <string name="122c2">Atenção à infertilidade; Contracepção clínica; Laqueadura; Vasectomia</string>
1134
1135 <string name="123cl">Serviço de Atenção ao Paciente com Tuberculose</string>
1136 <string name="123c2">Diagnóstico e tratamento Tratamento multidrogas resistentes</string>
1137
1138 <string name="124cl">Serviço de Atenção ao Pré-Natal, Parto e Nascimento</string>
1139 <string name="124c2">Acompanhamento do pré-natal de risco habitual; Acompanhamento do pré-natal de alto risco;
1140 Parto em gestação de risco habitual; Parto em gestação de alto risco; Centro de parto normal; Casa da Gestante, Bebê
1141 e Puérpera</string>
1142
1143 <string name="125cl">Serviço de Atenção Cardiovascular/Cardiologia</string>
1144 <string name="125c2">Eletrofisiologia; Cirurgia cardiovascular (adulto); Cirurgia cardiovascular (Pediátrico);
1145 Cirurgia vascular; Cardiologia intervencionista (hemodinâmica); Cardiologia Endovascular Extracardiaco; Cardiologia
1146 Clínica</string>
1147
1148 <string name="126cl">Serviço de Atenção Domiciliar</string>
1149 <string name="126c2">Assistência Domiciliar; Internação Domiciliar; Equipe Multidisciplinar de Atenção Domiciliar
1150 - EMAD; Equipe Multidisciplinar de Apoio - EMAP</string>
1151
1152 <string name="127cl">Serviço de Atenção em Neurologia/ Neurocirurgia</string>
1153 <string name="127c2">Neurocirurgia do trauma e anomalias do desenvolvimento; Coluna e nervos periféricos; Tumores
1154 do sistema nervoso; Neurocirurgia vascular; Tratamento neurocirúrgico da dor funcional; Investigação e cirurgia de
1155 epilepsia; Tratamento endovascular; Neurocirurgia funcional estereotáxica; Polissonografia</string>
1156
1157 <string name="128cl">Serviço de Atenção em Saúde Bucal</string>
1158 <string name="128c2">Dentística; Endodontia; Periodontia clínica; Moldagem/Manutenção; Cirurgia oral; Cirurgia
1159 buco-maxilo-facial; Atendimento à pessoa com deficiência</string>
1160

```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
1153 <string name="129c1">Serviço de Atenção em Urologia</string>
1154 <string name="129c2">Urologia em Geral; Litotripsia</string>
1155
1156 <string name="130c1">Serviço de Atenção Integral em Hanseníase</string>
1157 <string name="130c2">Serviço de Atenção Integral em Hanseníase Tipo I, Serviço de Atenção Integral em Hanseníase
Tipo II, Serviço de Atenção Integral em Hanseníase Tipo III</string>
1158
1159 <string name="131c1">Serviço de Atenção Psicossocial</string>
1160 <string name="131c2">Residência terapêutica em saúde mental; Atendimento psicossocial; Serviço Hospitalar para
Atenção a Saúde Mental; Serviço Residencial Terapêutico Tipo I - SRT tipo I; Serviço Residencial Terapêutico Tipo II
- SRT tipo II; UA Adulto; UA Infantojuvenil; Unidade de Atenção em Regime Residencial</string>
1161
1162 <string name="132c1">Serviço de Atendimento Móvel de Urgências</string>
1163 <string name="132c2">Ambulância de transporte; Unidade de suporte básico de vida terrestre (USB); Unidade de
suporte avançado de vida terrestre (USA); Unidade de Suporte Básico de vida: Equipe Embarcação; Veículos de
intervenção rápida; Outros veículos; Ambulância de resgate; Motolância; Unidade de Suporte Avançado de vida: Equipe
Embarcação; Equipe Aeromédico</string>
1164
1165 <string name="133c1">Serviço de Banco de Tecidos</string>
1166 <string name="133c2">Banco de valvula cardiaca humana; Banco de tecido musculo esquelético</string>
1167
1168 <string name="134c1">Serviço de Cirurgia Reparadora</string>
1169 <string name="134c2">Tratamento da lipodistrofia do portador de HIV / AIDS; Tratamento em queimados; Tratamento
da lipoatrofia facial do portador de HIV / AIDS</string>
1170
1171 <string name="135c1">Serviço de Cirurgia Reparadora</string>
1172 <string name="135c2">Cirurgia torácica</string>
1173
1174 <string name="136c1">Serviço de Controle de Tabagismo</string>
1175 <string name="136c2">Abordagem e tratamento do fumante</string>
1176
1177 <string name="137c1">Serviço de Cuidados Intermediários</string>
1178 <string name="137c2">NEONATAL CONVENCIONAL; NEONATAL CANGURU; PEDIÁTRICO; ADULTO</string>
1179
1180 <string name="138c1">Serviço de Diagnóstico por Anatomia Patológica e/ou Citopatologia</string>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
1181     <string name="I38c2">Exames anatomopatológicos; Exames citopatológicos; Monitoramento Externo da Qualidade dos
Exames Citopatológicos do Colo do Útero</string>
1182
1183     <string name="I39c1">Serviço de Diagnóstico por Laboratório Clínico</string>
1184     <string name="I39c2">Exames Bioquímicos; Exames hematológicos e hemostasia; Exames sorológicos e imunológicos;
Exames coprológicos; Exames de uroanálise; Exames hormonais; Exames de vigilância epidemiológica e ambiental;
Exames toxicológicos ou de monitorização terapêutica; Exames microbiológicos; Exames em outros líquidos biológicos;
Exames de genética; Exames para triagem neonatal; Exames imunohematológicos</string>
1185
1186     <string name="I40c1">Serviço de Diagnóstico por Métodos Gráficos/Dinâmicos</string>
1187     <string name="I40c2">Teste Ergométrico; Teste de Holter; Exame Eletrocardiográfico; Exame Eletroencefalográfico;
Teste Ergométrico por Telemedicina; Teste Holter por Telemedicina; Exame Eletrocardiográfico por Telemedicina; Exame
Eletroencefalográfico por Telemedicina; Exame Eletroneuromiografia; Exame videoeletroencefalografia; Potenciais
Evocados; Potenciais Evocados por Telemedicina</string>
1188
1189     <string name="I41c1">Serviço de Dispensação de Órteses, Próteses e Materiais Especiais</string>
1190     <string name="I41c2">OPM auxiliares de locomoção; OPM ortopédicas; OPM auditivas; OPM oftalmológicas; OPM em
gastroenterologia; OPM em urologia; OPM em odontologia; OPM buco-maxilo-facial; Substituição/Troca em órteses/
próteses; OPM em queimados; OPM em nefrologia; OPM - Projeto Olhar Brasil; OPM Projeto Olhar Brasil - Programa Saúde
na Escola (PSE); OPM Projeto Olhar Brasil - Programa Brasil Alfabetizado (PBA)</string>
1191
1192     <string name="I42c1">Serviço de Endocrinologia</string>
1193     <string name="I42c2">Diagnóstico/Tratamento das doenças endócrinas, metabólicas e nutricionais; Cirurgia de
glândulas endócrinas</string>
1194
1195     <string name="I43c1">Serviço de Endoscopia</string>
1196     <string name="I43c2">Do aparelho digestivo; Do aparelho respiratório; Do aparelho urinário; Do aparelho
ginecológico</string>
1197
1198     <string name="I44c1">Serviço de Farmácia</string>
1199     <string name="I44c2">Dispensação de Medicamentos do Componente Especializado da Assistência Farmaceutica;
Farmácia Popular; Farmácia com manipulação homeopática; Dispensação de medicamentos estratégicos; Dispensação de
medicamentos básicos; Farmácia Hospitalar</string>
1200
1201     <string name="I45c1">Serviço de Fisioterapia</string>
```


File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
1202     <string name="I45c2">Assistência fisioterapêutica em alterações obstétricas, neonatais e uroginecológicas;  
Assistência fisioterapêutica em alterações oncológicas; Assistência fisioterapêutica em oftalmologia; Assistência  
fisioterapêutica cardiovasculares e pneumo-funcionais; Assistência fisioterapêutica nas disfunções músculo-  
esqueléticas (todas as origens); Assistência fisioterapêutica em queimados; Assistência fisioterapêutica nas  
alterações em neurologia; Diagnóstico cinético funcional</string>  
1203  
1204     <string name="I46c1">Serviço de Hemoterapia</string>  
1205     <string name="I46c2">Procedimentos destinados à obtenção do sangue para fins de assistência hemoterápica;  
Diagnóstico em hemoterapia; Procedimentos especiais em hemoterapia; Medicina transfusional</string>  
1206  
1207     <string name="I47c1">Serviço de Laboratório de Histocompatibilidade</string>  
1208     <string name="I47c2">Exames de histocompatibilidade por meio de sorologia; Exames de histocompatibilidade por  
sorologia e biologia</string>  
1209  
1210     <string name="I48c1">Serviço de Laboratório de Prótese Dentaria</string>  
1211     <string name="I48c2">Laboratório Regional de Prótese Dentaria</string>  
1212  
1213     <string name="I49c1">Serviço de Oftalmologia</string>  
1214     <string name="I49c2">Diagnóstico em oftalmologia; Tratamento clínico do aparelho da visão; Tratamento cirúrgico  
do aparelho da visão; Projeto Olhar Brasil; Diagnóstico em Oftalmologia por Telemedicina; Projeto Olhar Brasil -  
Programa Saúde na Escola (PSE); Projeto Olhar Brasil - Programa Brasil Alfabetizado (PBA)</string>  
1215  
1216     <string name="I50c1">Serviço de Oncologia</string>  
1217     <string name="I50c2">Oncologia pediátrica; Hematologia; Oncologia clínica; Radioterapia; Oncologia cirúrgica</  
string>  
1218  
1219     <string name="I51c1">Serviço de Órteses, Próteses e Materiais Especiais em Reabilitação</string>  
1220     <string name="I51c2">Dispensação de OPM Auxiliares de Locomoção; Manutenção e Adaptação de OPM Auxiliares de  
Locomoção; Dispensação de OPM Ortopédica; Manutenção e Adaptação de OPM Ortopédica; Dispensação de OPM Auditiva;  
Manutenção e Adaptação de OPM Auditiva; Dispensação de OPM Oftalmológica; Manutenção e Adaptação de OPM  
Oftalmológica; Substituição/Troca de OPM</string>  
1221  
1222     <string name="I52c1">Serviço de Pneumologia</string>  
1223     <string name="I52c2">Tratamento de Doenças das Vias Aéreas Inferiores; Diagnóstico em Pneumologia; Diagnóstico  
em Pneumologia por Telemedicina</string>
```

```

1224
1225     <string name="I53c1">Serviço de Práticas Integrativas e Complementares</string>
1226     <string name="I53c2">Acupuntura; Fitoterapia; Outras técnicas em medicina tradicional chinesa; Práticas corporais
    /Atividade física; Homeopatia; Termalismo/Crenoterapia; Medicina antroposófica</string>
1227
1228     <string name="I54c1">Serviço de Reabilitação</string>
1229     <string name="I54c2">Reabilitação Visual; Reabilitação Intelectual; Reabilitação Física; Reabilitação visual/
    mental/múltiplas deficiências; Reabilitação Auditiva; Oficina Ortopédica Fixa; Oficina Ortopédica Intinerante
    Terrestre; Oficina Ortopédica; Intinerante Fluvial; Atenção Fonoaudiológica; Atenção Fisioterapeutica; Atenção à
    Saúde das Pessoas Ostomizadas I; Atenção à Saúde das Pessoas Ostomizadas II</string>
1230
1231     <string name="I55c1">Serviço de Suporte Nutricional</string>
1232     <string name="I55c2">Enteral; Enteral/Parenteral; Enteral/Parenteral, com manipulação/fabricação</string>
1233
1234     <string name="I56c1">Serviço de Telemedicina</string>
1235     <string name="I56c2">Exame eletrocardiográfico por telemedicina</string>
1236
1237     <string name="I57c1">Serviço de Terapia Intensiva</string>
1238     <string name="I57c2">ADULTO; NEONATAL; PEDIÁTRICO; QUEIMADOS; DOENÇA CORONARIANA (UCO)</string>
1239
1240     <string name="I58c1">Serviço de Traumatologia e Ortopedia</string>
1241     <string name="I58c2">Serviço de Traumatologia e Ortopedia; Serviço de Traumatologia e Ortopedia Pediátrica (até
    21 anos de idade); Serviço de Traumatologia e Ortopedia de Urgência</string>
1242
1243     <string name="I59c1">Serviço de Traumato-Ortopedia</string>
1244     <string name="I59c2">Diagnóstico e tratamento em cintura escapular; Diagnóstico e tratamento em membros
    superiores; Diagnóstico e tratamento em coluna vertebral e caixa torácica; Diagnóstico e tratamento em cintura
    pélvica; Diagnóstico e tratamento em membros inferiores</string>
1245
1246     <string name="I60c1">Serviço de Triagem Neonatal</string>
1247     <string name="I60c2">Tratamento de recém-nascido com hipotireoidismo e fenilcetonuria; Tratamento de recém-
    nascido com doenças falciformes; Tratamento de recém-nascido com fibrose cística; Tratamento de recém-nascido com
    outras doenças congênitas; Acompanhamento a pacientes com diagnósticos de Hiperplasia Adrenal Congênita;
    Acompanhamento a pacientes com diagnósticos de Deficiência Biotinidase</string>
1248

```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\strings.xml

```
1249     <string name="l61cl">Serviço de Urgência e Emergência</string>
1250     <string name="l61c2">Pronto Socorro Especializado; Pronto Atendimento; Estabilização de
    paciente crítico/grave; Atendimento ao paciente com Acidente Vascular Cerebral (AVC); Pronto Atendimento Clínico;
    Pronto Atendimento Pediátrico; Pronto Atendimento Obstétrico; Pronto Atendimento Psiquiátrico; Pronto Atendimento
    Oftalmológico; Pronto Atendimento Odontológico; Pronto Socorro Pediátrico; Pronto Socorro Obstétrico; Pronto Socorro
    Cardiovascular; Pronto Socorro Neurologia/Neurocirurgia; Pronto Socorro Traumatológico; Pronto Socorro
    Odontológico; Pronto Socorro Oftalmológico; Pronto Socorro Geral/Clínico</string>
1251
1252     <string name="l62cl">Serviço de Videolaparoscopia</string>
1253     <string name="l62c2">Diagnóstica; Cirúrgica</string>
1254
1255     <string name="l63cl">Serviço de Vigilância em Saúde</string>
1256     <string name="l63c2">Vigilância epidemiológica; Vigilância sanitária; Vigilância ambiental; Núcleo de vigilância
    hospitalar</string>
1257
1258     <string name="l64cl">Serviço Móvel de Urgência (exceto SAMU)</string>
1259     <string name="l64c2">Transporte terrestre; Transporte aéreo; Transporte marítimo</string>
1260
1261     <string name="l65cl">Serviço Posto de Coleta de Materiais Biológicos</string>
1262     <string name="l65c2">Coleta realizada fora das estrutura laboratorial</string>
1263
1264     <string name="l66cl">Teleconsultoria</string>
1265     <string name="l66c2">Teleconsultoria Assíncrona; Teleconsultoria Síncrona; Segunda Opinião Formativa</string>
1266
1267     <string name="l67cl">Transplante</string>
1268     <string name="l67c2">Rim; Medula óssea; Coração; Pulmão; Córnea / Esclera; Fígado; Pâncreas; Retirada de órgãos;
    Pele; Válvulas cardíacas; Osteocondrologamentos; Separação, avaliação biomicroscópica e conservação da córnea/esclera
    ; Contagem endotelial corneana; Acompanhamento de paciente transplantado; Ações para doação e captação de órgãos e
    tecidos; Retirada de globo ocular humano para transplante</string>
1269     <string name="msg_info preferencias">Em sua primeira visualização do aplicativo foi utilizada a sua posição
    geográfica atual para definir o município onde você está localizado neste momento. É interessante que você selecione
    na lista acima outros municípios de seu interesse para que apareçam nos resultados de suas buscas por
    estabelecimentos.</string>
1270
1271 </resources>
```

File - E:\Android\Projetos\CadEsaude\app\src\main\res\values\styles.xml

```
1 <resources>
2
3     <style name="MeuTemaMaterial" parent="MeuTemaMaterial.Base"></style>
4
5     <style name="MeuTemaMaterial.Base" parent="Theme.AppCompat.Light.DarkActionBar">
6         <item name="windowNoTitle">true</item>
7         <item name="windowActionBar">false</item>
8         <item name="colorPrimary">@color/primary</item>
9         <item name="colorPrimaryDark">@color/primary_dark</item>
10        <item name="colorAccent">@color/accent</item>
11        <item name="android:textColorPrimary">@color/primary_text</item>
12        <item name="android:textColorPrimaryInverse">@color/primary_text</item>
13        <item name="android:textColorSecondary">@color/secondary_text</item>
14    </style>
15
16    <style name="MeuTemaMaterial.Prefencias" parent="MeuTemaMaterial">
17        <item name="android:actionBarTheme">@style/ThemeOverlay.AppCompat.Dark.ActionBar</item>
18        <item name="windowActionBar">true</item>
19        <item name="windowNoTitle">false</item>
20    </style>
21
22    <style name="MeuTemaMaterial.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar" />
23
24    <style name="MeuTemaMaterial.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />
25
26    <style name="FullscreenTheme" parent="MeuTemaMaterial">
27        <item name="android:actionBarStyle">@style/FullscreenActionBarStyle</item>
28        <item name="android:windowActionBarOverlay">true</item>
29        <item name="android:windowBackground">@null</item>
30        <item name="metaButtonBarStyle">?android:attr/buttonBarStyle</item>
31        <item name="metaButtonBarButtonStyle">?android:attr/buttonBarButtonStyle</item>
32    </style>
33
34    <style name="FullscreenActionBarStyle" parent="Widget.AppCompat.ActionBar">
35        <item name="android:background">@color/black_overlay</item>
36    </style>
```

File - E:\Android\Projetos\CadESaude\app\src\main\res\values\styles.xml

36 </style>

37

38 </resources>

39

File - E:\Android\Projetos\CadEsaude\app\src\main\res\xml\preferencias.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
3     android:title="@string/titulo_config">
4
5     <PreferenceCategory android:title="Municípios">
6         <MultiSelectListPreference
7             android:defaultValue="@array/municipios_preferidos_default"
8             android:dialogTitle="@string/municipios_dialog_title"
9             android:entries="@array/nomes_municipios_view"
10            android:entryValues="@array/codigos_municipios"
11            android:key="@string/pref_municipios_preferidos_key"
12            android:selectable="true"
13            android:summary="@string/municipios_preferidos_subtitulo"
14            android:title="@string/municipios_preferidos_titulo" />
15        </PreferenceCategory>
16
17        <PreferenceCategory android:title="Estabelecimentos">
18            <br.com.jucemar_dimon.cadesaude.SeekBarPreference
19                android:defaultValue="1"
20                android:key="@string/pref_distancia_estabelecimentos_key" />
21        </PreferenceCategory>
22
23        <br.com.jucemar_dimon.cadesaude.InfoPreference
24            android:enabled="false"
25            android:key="@string/pref_info_key"
26            android:selectable="false" />
27    </PreferenceScreen>
28
29

```

File - E:\Android\Projetos\CadESaude\app\src\main\res\xml\searchable.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <searchable xmlns:android="http://schemas.android.com/apk/res/android"
3     android:hint="Pesquisa"
4     android:label="@string/app_name">
5 </searchable>
```

File - E:\Android\Projetos\CadESaude\app\build.gradle

```

1 apply plugin: 'com.android.application'
2
3 android {
4     compileSdkVersion 23
5     buildToolsVersion "23.0.1"
6
7     defaultConfig {
8         applicationId "br.com.jucemar_dimon.cadesaude"
9         minSdkVersion 14
10        targetSdkVersion 23
11        versionCode 12
12        versionName "1.0.1"
13    }
14    buildTypes {
15        release {
16            minifyEnabled false
17            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
18        }
19    }
20 }
21
22 dependencies {
23     compile fileTree(include: ['*.jar'], dir: 'libs')
24     testCompile 'junit:junit:+'
25
26     compile 'com.android.support:appcompat-v7:23.+'
27     compile 'com.android.support:design:23.+'
28     compile 'com.android.support:support-v4:23.+'
29     compile 'com.android.support:recyclerview-v7:23.+'
30     compile 'com.github.clans:fab:1.6.2'
31     compile 'com.google.android.gms:play-services-location:+'
32
33
34 }

```


APÊNDICE D - ARTIGO SOBRE O TCC

CadESaúde: Um aplicativo para a plataforma Android para localizar estabelecimentos de saúde públicos e privados no Estado de Santa Catarina

Jucemar Dimon¹

¹Curso de Sistemas de Informações – Instituto de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC) Florianópolis – SC – Brasil

jucemar.dimon@grad.ufsc.br

Abstract. *There are several ways to locate health information on the internet, but they can come from many different sources and may be outdated. Differently from the search results on the Internet, the National Registry of Health Establishments (CNES) is the official data source of the Ministry of Health. The system developed in this work provided a solution to the population of the State of Santa Catarina, which assists in the search of establishments of health. The application developed for the Android platform was distributed in the Google Play Store and provides the updated data about these establishments, because it uses the same information contained in CNES in its internal database.*

Resumo. *Há na internet diversas formas de se localizar informações de um estabelecimento de saúde, porém podem vir de muitas fontes diferentes e podem estar desatualizadas. Diferentemente dos resultados de busca na internet, o Cadastro Nacional de Estabelecimentos de Saúde (CNES) é a fonte de dados oficial do Ministério da Saúde. O sistema desenvolvido neste trabalho forneceu uma solução à população do Estado de Santa Catarina que auxilia na busca de estabelecimentos de saúde. O aplicativo desenvolvido para a plataforma Android foi distribuído na Google Play Store e fornece os dados devidamente atualizados sobre esses estabelecimentos, pois utiliza em seu banco de dados interno as mesmas informações contidas no CNES.*

1. Introdução

Com a grande difusão e popularização de sites de anúncios para as empresas divulgarem seus serviços na internet, muitas pessoas podem preferir esses meios para encontrarem a informação que necessitam sobre uma empresa. Ao realizar uma busca por um tipo de estabelecimento específico, como uma clínica médica, hospital ou posto de saúde utilizando esta metodologia, recebe-se uma grande quantidade de sites com informações sobre o local desejado como: MAPLink⁶⁸; GuiaMais⁶⁹; Telelistas⁷⁰; Foursquare⁷¹ e Google Maps⁷². Esses permitem que usuários terceiros cadastrem informações sobre as empresas, no entanto, posteriormente, tais

⁶⁸ <http://maplink.com.br/>

⁶⁹ <http://www.guiamais.com.br/florianopolis-sc>

⁷⁰ <http://www.telelistas.net/>

⁷¹ <https://pt.foursquare.com/>

⁷² <https://www.google.com.br/maps>

informações podem estar desatualizadas e podem prejudicar quem necessita de sua localização de forma rápida.

O avanço das plataformas mobile e a sua popularização no mercado de smartphones, que em 2014 teve um crescimento de 23% segundo artigo publicado por Exame.com (2015), vêm despertando grande interesse nos desenvolvedores de aplicações para esta plataforma. Nesse sentido, muitos estabelecimentos de saúde como a Clínica Privada de Especialidades Villa María (2015) já possuem seu aplicativo para divulgação de seus serviços e operadoras de convênios médicos, como Portal Unimed Ltda (2015), sendo disponibilizado como ferramenta para seus clientes localizarem seus profissionais credenciados. As empresas do ramo de tecnologia que desenvolvem esses aplicativos, também projetam soluções genéricas para auxiliar os usuários de smartphones na localização de uma clínica, hospital, posto de saúde, entre outros, como desenvolvido por Wintimeipc (2015) em seu aplicativo MedSquare para a plataforma Android.

Pode-se perceber que existem várias formas de se localizar um estabelecimento de saúde: através de provedores de busca na internet; através de aplicativos específicos de operadoras de convênios de saúde; através de aplicativos dos próprios estabelecimentos e, até mesmo, através de catálogos impressos em papel. São variadas fontes de informações diferentes e separadas uma das outras, o que pode causar divergências nos dados além de poder prejudicar o cidadão que necessita de um rápido resultado. Foi pensando neste contexto que se desenvolveu este trabalho, com o objetivo de criar uma solução que pudesse auxiliar o cidadão a localizar um estabelecimento de saúde, o qual atenda suas necessidades de forma eficiente. Utilizando-se de tecnologias atuais da plataforma mobile como o Android, desenvolveu-se um aplicativo off-line que utiliza como banco de dados as informações contidas no CNES (Cadastro Nacional dos Estabelecimentos de Saúde), optou-se por este por serem informações fidedignas dos estabelecimentos de saúde, sendo a única fonte de consulta utilizada pelo aplicativo. Dessa forma, torna-se possível que os usuários utilizem um único aplicativo para realizar as buscas por estabelecimentos de saúde de todos os tipos, sendo eles públicos ou privados.

2. Conceitos

Nesta seção serão levantados, de forma sucinta, alguns conceitos que ajudarão na compreensão do tema focado neste artigo.

2.1. Webcrawler

Um web crawler, segundo Tavares e Lóscio (2013), é um agente de software que acessa a WEB de forma automatizada através de URLs para extrair dados, sendo esse processo denominado crawling. O trabalho de Tavares, Lóscio e Oliveira (2012) complementa o conceito definindo que este componente automatizado visita as páginas HTML e navega pelo seu conteúdo capturando-o, os links contidos nas páginas são adicionados a sua fila de URLs que serão visitadas na sequência, assim, um web crawler pode varrer toda a web de forma recursiva.

2.2. Data Access Object

O padrão de desenvolvimentos conhecido como DAO, do inglês *Data Access Object*, segundo Taborda (2015), é um padrão introduzido no ambiente JEE para simplificar e desacoplar a interação das aplicações Java com a API JDBC. No padrão DAO, todas as comunicações com o mecanismo de persistência são mediadas por um objeto escrito especificamente para mapear as informações dos objetos específicos de transporte para a API de persistência e vice e versa. Toda a lógica de mapeamento e execução das instruções é deixada dentro do objeto DAO, desta

forma, isolando a aplicação da API de persistência por completo. O objeto DAO é criado para operar e interagir com mecanismo de persistência em nome da aplicação.

2.3. Singleton

O Singleton é uma convenção que garante que um único objeto de uma classe de interesse será instanciado e que há apenas um ponto global de acesso a ela, ou seja, em qualquer ponto da aplicação de uma forma geral caso haja a intenção de uso do objeto cuja classe foi implementada usando este padrão, chama-se o construtor do objeto e caso ele já tenha sido criado, tem-se uma referência dele. Segundo Freeman et al. (2009), este tipo de padrão de projeto é muito útil quando os objetos possuem atributos de configuração que não podem se repetir.

2.4. Cadastro Nacional dos Estabelecimentos de Saúde

O CNES (Cadastro Nacional dos Estabelecimentos de Saúde) é um projeto do Ministério da Saúde que consiste na criação de um banco de dados unificado, contendo o cadastro detalhado de todos os estabelecimentos de saúde do país que prestam atendimento à população. Este projeto foi instituído com a publicação da Portaria nº 511 de Brasil (2000), a qual detalha como deve ser feito todo o processo de cadastramento, atualização e transmissão dos dados de todos os estabelecimentos do país para o banco de dados central do Ministério da Saúde, administrado pelo DATASUS. Ao longo dos anos diversas portarias, notas técnicas e manuais tem sido publicado para regular os processos dos responsáveis pela realização do cadastro e atualização dessas informações, como é o caso da Portaria nº 1.646 de Brasil (2015), que reitera a importância e obrigatoriedade do cadastramento de todos os estabelecimentos de saúde pelos responsáveis de cada município, de sua manutenção e veracidade dos dados que são alimentados no sistema, assim como da transmissão mensal dessas informações do banco de dados local para o banco de dados federal, com a possibilidade de aplicação de penalidade aos gestores dos municípios caso essas exigências não sejam atendidas.

Os responsáveis pelo cadastro dos estabelecimentos são as Secretarias Municipais ou Estaduais de Saúde, esse trata-se do preenchimento de uma série de fichas que detalham todas as características físicas, jurídicas e estruturais do local onde funciona o estabelecimento, assim como os recursos humanos, tecnológicos e serviços oferecidos. Esses dados são adicionados a um banco de dados local através de um software específico, o qual na sequência pode conectar-se ao servidor central federal ou estadual para sincronização dos dados.

2.5. Android

O Android é um sistema operacional para dispositivos móveis, sendo a base do mesmo originada em 2003, na cidade de Palo Alto na Califórnia. Segundo Meyer (2016), ele foi desenvolvido por Andy Rubin, Rich Miner, Nick Sears e Chris White, tais empresários já iniciados no ramo da tecnologia, foram os fundadores da Android, Inc.. Em 2005 a Google adquiriu a empresa Android, Inc, e criou a sua nova divisão de pesquisa em tecnologia móvel, mas foi somente a partir de 2007 que um grande passo foi dado, marcando a história do Android e fazendo com que outras empresas da mesma divisão de mercado comesçassem a se preocupar: grandes fabricantes como Samsung, Sony, HTC, operadoras como as americanas Sprint Nextel e T-Mobile, e fabricantes de hardware como Qualcomm e a Texas Instrumentes, além do próprio Google, reuniram-se em um consórcio de tecnologia e fundaram a OHA⁷³. Várias outras grandes empresas do ramo de telefonia, hardware, semicondutores e de marketing e comércio participam e contribuem para a sua evolução, sendo que todas essas empresas compartilham suas expectativas e *insights* nos congressos promovidos pela OHA(Open Handset Alliance). O

⁷³ Open Handset Alliance - <http://www.openhandsetalliance.com>

principal objetivo deste consórcio é a busca de um padrão de dispositivo móvel aberto e que atenda às necessidades atuais de todos.

De acordo com Alves (2016), a Google ao longo dos anos vem evoluindo seu sistema operacional de forma cada vez mais rápida. Lançado em 2009, o Android 1.5 (*Cupcake*) foi sua versão mais rudimentar, porém, ainda no mesmo ano, recebeu duas novas versões, respectivamente: Android 1.6 (*Donut*) e Android 2.0 (*Eclair*). No ano de 2010, a Google lança Android 2.2 (*Froyo*), além disso, no final do mesmo ano é liberado o Android 2.3 (*Gingerbread*), tornando-se uma das melhores versões já lançadas até aquele momento pela Google, esta liderou a popularidade por um longo tempo entre os aparelhos da época. No início de 2011, a versão menos lembrada do Android foi lançada, o Android 3.0 (*Honeycomb*), voltada especificamente para *tablets*, já no fim de 2011 o Google lançou a versão 4.0 (*Ice Cream Sandwich*) baseada no padrão de *design* do Android 3.0. Em 2012, é lançado o Android 4.1 (*Jelly Bean*) e suas atualizações que se estenderam da 4.1 até a 4.3. No ano de 2013, é lançado o Android 4.4 (*Kitkat*). Uma revolução visual foi promovida pelo lançamento do Android 5.0 (*Lollipop*) no fim de 2014, com sua linguagem visual denominada *Material Design*. No segundo semestre de 2015, o Android 6.0 (*Marshmallow*) foi apresentado oficialmente trazendo revoluções nas questões de segurança e consumo de bateria.

Em sua arquitetura, que pode ser vista na Figura 1, percebe-se que o sistema tem como base um kernel Linux. Este kernel faz toda a comunicação com a camada de hardware através de seus drivers.

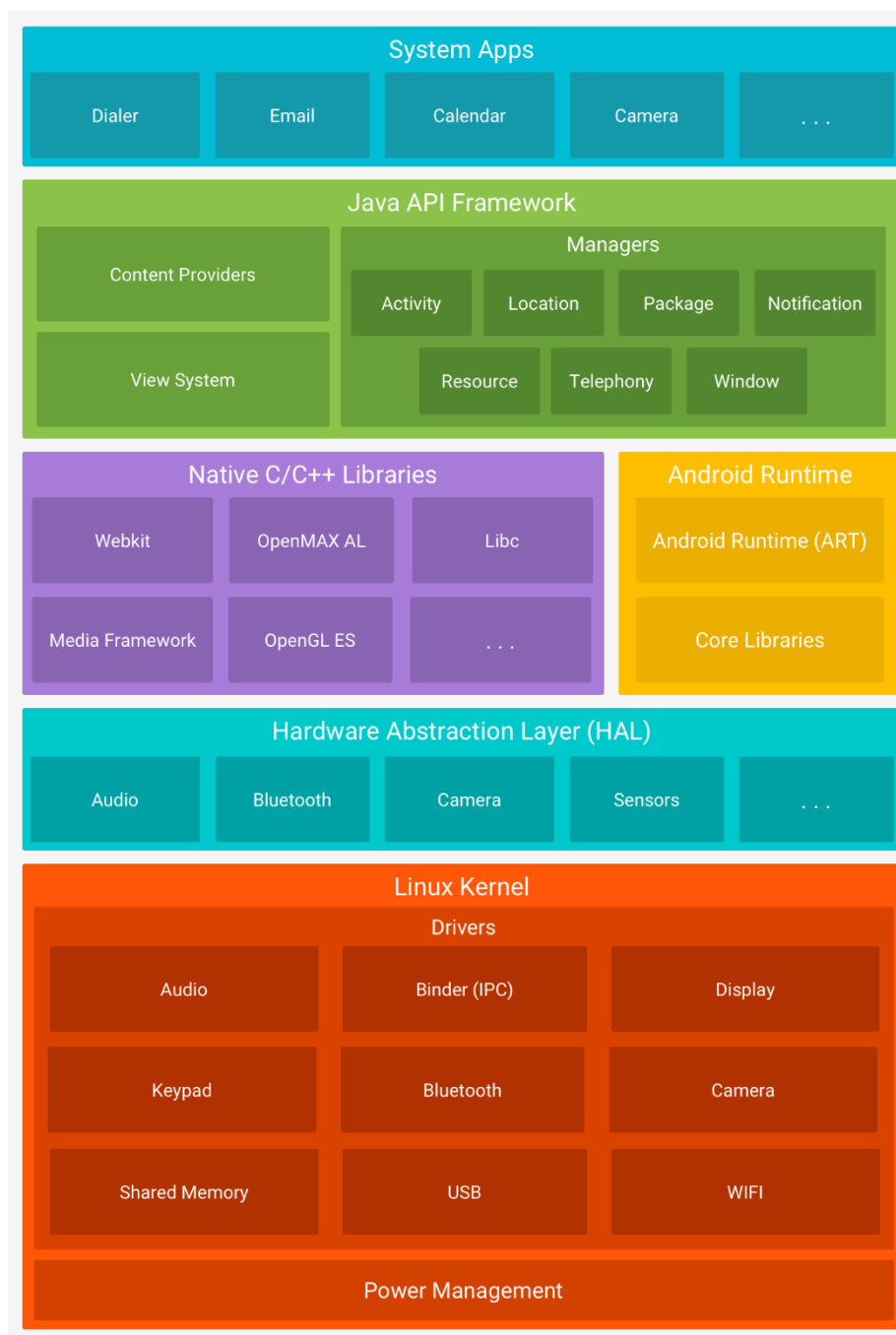


Figura 42. Arquitetura da plataforma Android

Fonte: Android (2016)

Trabalhando em conjunto com o *kernel* do Linux existe a HAL⁷⁴ (*Hardware Abstraction Layer*). Esta camada é composta por vários módulos e interfaces que fornecem a capacidade de interação com uma linguagem de alto nível como o Java. Ao lado existe na estrutura da plataforma o *Android Runtime* que é composto pelas *Core Libraries*, juntamente com a máquina virtual do Android. As *Core Libraries* são as responsáveis por disponibilizar ao desenvolvedor tudo que é utilizável pelo Java SE. A ART (*Android Runtime*) é a máquina virtual do Android utilizada para a execução dos aplicativos. Como os aplicativos são desenvolvidos e compilados em Java eles precisam de uma máquina virtual para poderem ser

⁷⁴ <https://source.android.com/devices/index.html#Hardware Abstraction Layer>

interpretados. Acima dessas camadas existe o *framework* de aplicações do Android, neste estão as funcionalidades que vão permitir ao desenvolvedor criar seus aplicativos. No topo desta estrutura temos a camada de aplicações onde ficam as aplicações do Android. Por padrão o sistema operacional já possui diversos aplicativos nativo, porém como os aplicativos de terceiros também serão posicionados nessa camada, é possível substituir um aplicativo nativo por um terceiro, pois a plataforma percebe ambos da mesma forma.

3. Trabalhos correlatos

No ano de 2014, os autores Sarrinho e Campos (2014) publicaram um artigo relatando o desenvolvimento do projeto SAMob (Sistema Móvel de Geolocalização e Geoprocessamento para Locais de Atendimento em Saúde). O sistema tinha como funcionalidades a possibilidade de seus usuários localizarem um estabelecimento de saúde e além disso, compartilhar, avaliar suas opiniões sobre o mesmo. O sistema era composto por duas aplicações: uma servidora que era armazenada em nuvem e utilizava a plataforma da *Google App Engine* e outra cliente através de um aplicativo para a plataforma Android. A aplicação servidora tinha como funcionalidade gerenciar os estabelecimentos de saúde e fornecer os dados as aplicações clientes que se conectavam, sendo esta última, um aplicativo que fazia a conexão com o servidor para retornar as informações sobre as buscas dos usuários, além disso, ela possibilitava aos usuários não somente a busca, mas também a qualificação dos serviços dos estabelecimentos através de comentários. Após a conclusão do desenvolvimento, os autores submeteram o sistema a um teste de usabilidade e obtiveram um bom resultado. Em suas conclusões do projeto, eles sugerem em seus trabalhos futuros que o sistema tenha seu aplicativo adaptado para outras plataformas móveis como: iPhone, Windows Phone e Firefox OS.

Nas lojas de aplicativos da Google e da IOS pode ser encontrado o aplicativo *MedSquare*, o qual foi desenvolvido pela empresa WINTIME IPC com o objetivo de melhorar o atendimento das pacientes nos estabelecimentos de saúde Medsquare (2015). Dentre as suas funcionalidades, os usuários podem: localizar estabelecimentos de saúde por nome, especialidade médica ou localização geográfica; comentar e qualificar sobre o atendimento naquele estabelecimento e interagir com outros usuários do aplicativo através de um chat. O aplicativo depende de uma conexão com a internet pois seus dados estão armazenados em um servidor, além disso, para que os estabelecimentos apareçam nos resultados de busca eles necessitam ser cadastrados no banco de dados, sendo que essa ação deve ser solicitada à empresa pelo responsável pelo estabelecimento. Este aplicativo já possui uma grande quantidade de usuários atualmente e a empresa investiu em atualizações de suas versões até 12 de agosto de 2015, data na qual a última atualização foi disponibilizada na loja de aplicativos da Google. Nesta última versão, aparece a informação de que a partir de agora o aplicativo possui em sua base de dados as informações do CNES. Não há nenhum tipo de informação referente a atualização desses dados, como são dados atualizados frequentemente pelo DATASUS, não há como saber se sua base de dados está atualizada.

4. O desenvolvimento do sistema

Nesta seção serão apresentadas as etapas que envolveram o desenvolvimento do projeto. Na primeira seção serão tratadas as questões relacionadas a ferramenta de captura de dados, nas próximas serão descritos todos os processos envolvidos no desenvolvimento do aplicativo para a plataforma Android.

4.1. A ferramenta de captura de dados

Para o desenvolvimento do aplicativo, primeiramente, foi necessário projetar quais informações sobre os estabelecimentos que seriam disponibilizadas aos usuários, sendo essas definidas como: Nome fantasia; Razão social; Endereço; Telefone; Profissionais que atuam no estabelecimento; Serviços; Categorias de atendimento; Convênios e coordenadas geográficas. Após a definição dos dados, um esquema de banco de dados foi criado para armazenar essas informações. O programa utilizado para criação do esquema foi o *DB Designer Fork*⁷⁵ e o esquema pode ser visto na Figura 2.

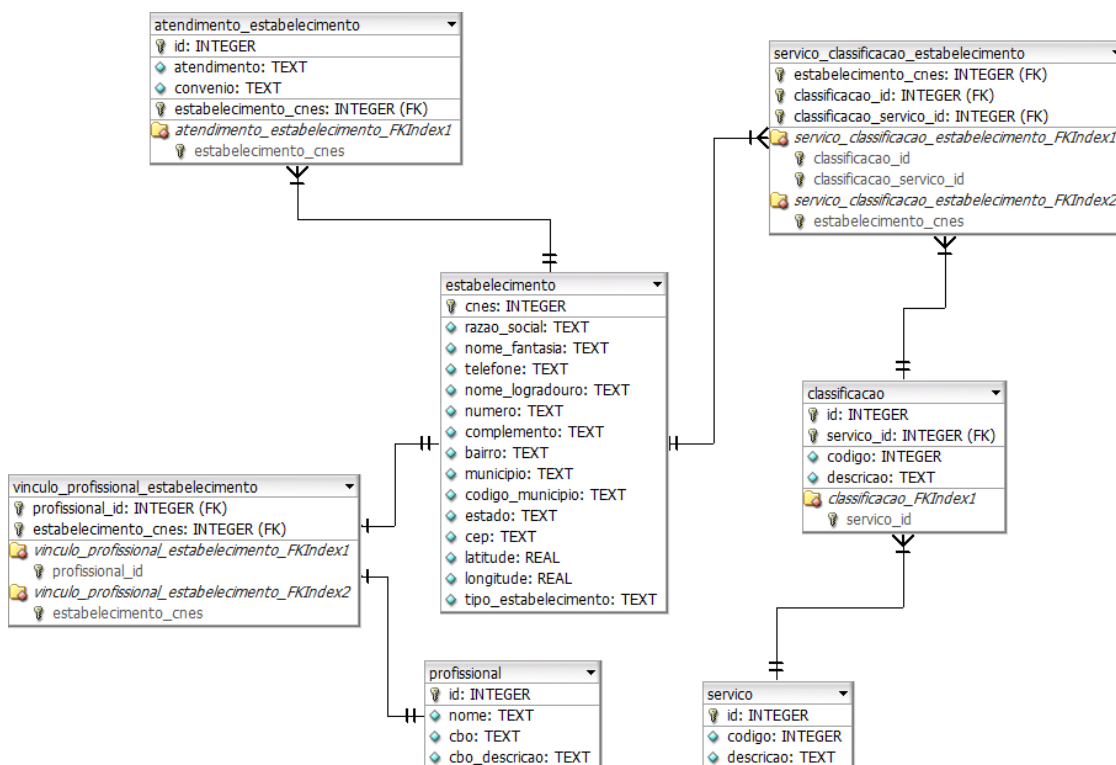


Figura 43. Esquema de bando de dados

Fonte: Desenvolvido pelo autor

A próxima etapa do desenvolvimento foi a criação de uma ferramenta que realizasse a captura dos dados do portal do CNES, essa foi desenvolvida na linguagem Java utilizando a IDE *Netbeans*⁷⁶ em sua versão 8. Para realizar tal ação a ferramenta deveria ter o comportamento de um *webcrawler*, varrendo as páginas HTML e capturando seu conteúdo que na sequencia seriam manipulados e extraídos os dados de interesse neste projeto. Para conseguir tal efeito na ferramenta foi utilizado a biblioteca para Java *Jsoup*⁷⁷ integrada ao projeto. Além dessa funcionalidade a ferramenta também necessitaria gerar e manipular um banco de dados para armazenar tais informações. O SGBD (Sistema Gerenciado de Banco de Dados) que foi utilizado nesta ferramenta foi o *SQLite*⁷⁸ em sua versão 3. A escolha desse SGBD foi tomada devido sua grande compatibilidade com a plataforma Android. Para que fosse possível trabalhar com este SGBD dentro da ferramenta foi utilizada a biblioteca para Java *sqlite-jdbc*⁷⁹. Com o objetivo de ilustrar as tarefas e o fluxo delas nesta aplicação de captura de dados, nomeada

⁷⁵ <http://sourceforge.net/projects/dbdesigner-fork>

⁷⁶ <https://netbeans.org>

⁷⁷ <https://jsoup.org>

⁷⁸ <https://sqlite.org>

⁷⁹ <https://bitbucket.org/xerial/sqlite-jdbc>

como *JWebCrawlerCNES*, foi criado um diagrama de atividades utilizando UML. Este diagrama pode ser visto em detalhes na Figura 3.

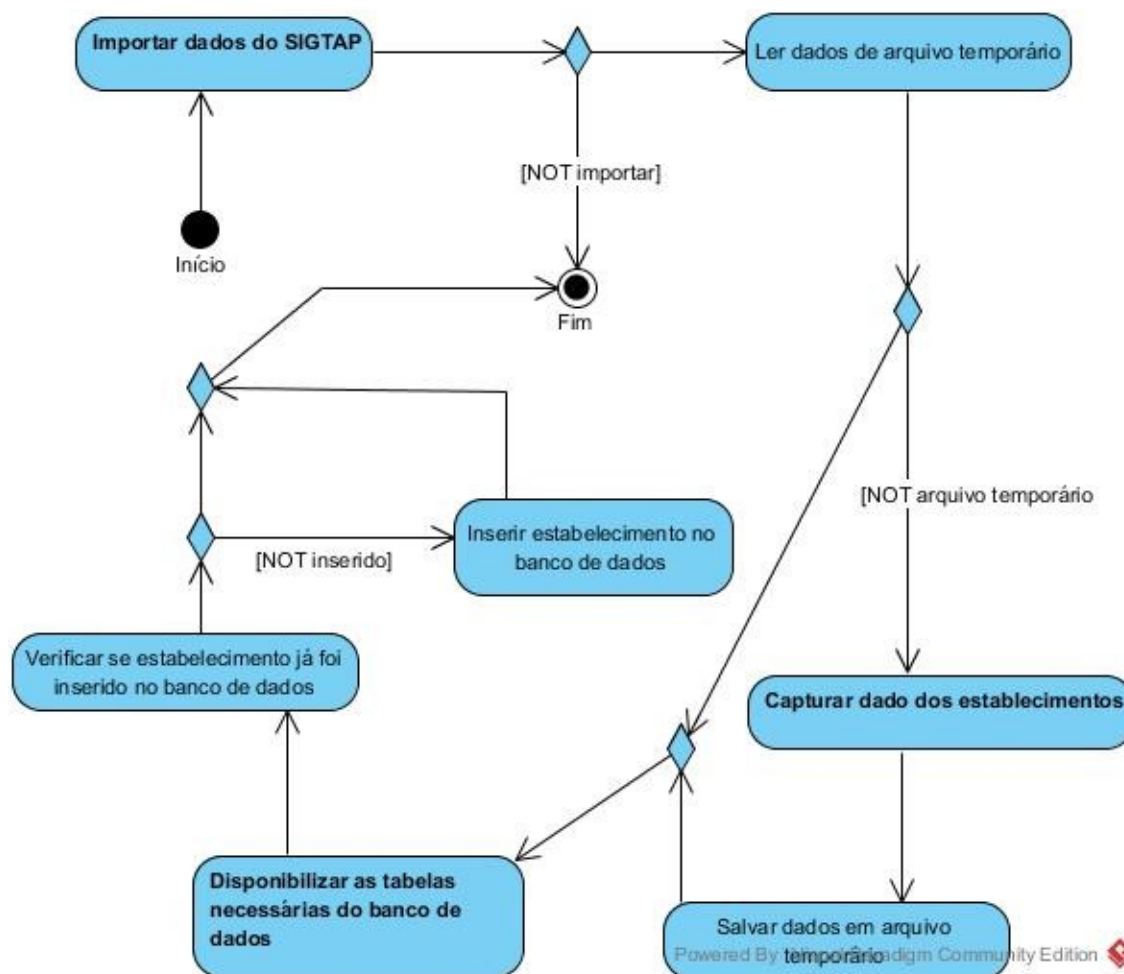


Figura 44. Diagrama de atividades UML

Fonte: Desenvolvido pelo autor

A primeira etapa realizada pela ferramenta é a captura dos dados referente ao SIGTAP (Sistema de Gerenciamento da Tabela de Procedimentos, Medicamentos e OPM do SUS). Estes dados, armazenados em arquivos de texto, são retirados do portal do DATASUS e importados pela ferramenta para preencher as tabelas *serviço* e *classificação* do banco de dados, para que se tenha a informação mais atualizada sobre estas informações. As coordenadas geográficas dos estabelecimentos são adquiridas dentro da atividade de capturar dados dos estabelecimentos mencionado pelo diagrama. Ao final do processo, a ferramenta gera um arquivo de banco de dados, sendo esse utilizado no aplicativo desenvolvido.

4.2. O aplicativo para Android

Após a execução da ferramenta de captura tem-se em mão um arquivo de banco de dados com informações de estabelecimentos de saúde de todos o Estado de Santa Catarina. A próxima ação foi definir a forma como esses dados seriam apresentados ao usuário através do aplicativo, então pensou-se nas funcionalidades que poderiam ser disponibilizadas aos usuários utilizando-se das possibilidades possíveis que um smartphone pode oferecer: realizar ligações telefônicas; possibilidade de trabalhar com geolocalização e a possibilidade de interagir com mapas. Dessa forma foram criadas as funcionalidades para o aplicativo. São elas:

- Realizar a busca por estabelecimentos nas redondezas;
- Filtrar os resultados das buscas por parâmetros;
- Visualizar todos os dados referentes a um estabelecimento;
- Realizar buscas por estabelecimentos dentro de um município utilizando parâmetros;
- Permitir ao usuário fazer uma ligação telefônica ao estabelecimento;
- Permitir ao usuário definir um estabelecimento como favorito;
- Permitir ao usuário visualizar a localização de um estabelecimento em um mapa.

Com as funcionalidades do aplicativo definidas, a próxima etapa foi a definição das telas que o usuário teria no aplicativo e a distribuição dessas funcionalidades entre as telas pensadas, sendo o fluxo do desenvolvimento baseado em telas. A cada definição de uma delas, o processo de programação foi iniciado, dessa forma, utilizou-se para tal a ferramenta *Android Studio*⁸⁰ em sua versão 2.2.2. Nas próximas seções serão apresentadas as telas desenvolvidas e suas funcionalidades.

4.2.1. A tela de apresentação do aplicativo

Ao abrir o aplicativo, o usuário será levado a tela de apresentação. Esta tela pode ser visualizada na Figura 4. Não há nenhuma funcionalidade em especial nela que o usuário possa realizar.



Figura 45. Tela de apresentação

Fonte: Desenvolvido pelo autor

Nesta tela de apresentação ocorre em segundo plano uma rotina de verificação para constatar se esta é a primeira vez que o aplicativo foi aberto, assim como se ele já foi configurado. Caso negativo, o usuário será levado a tela da Figura 5, na qual ele deve fazer a

⁸⁰ <https://developer.android.com/studio/index.html?hl=pt-br>

configuração inicial do aplicativo. No outro caso, o usuário será levado a tela da Figura 6, nessa ele poderá visualizar todos os seus estabelecimentos definidos como favoritos.

4.2.2. A tela de configuração inicial

A tela de configuração inicial tem como objetivo exigir do usuário que defina alguns parâmetros que são importantes para o funcionamento do aplicativo, na mesma os usuários devem definir qual o raio de distância em quilômetros que os seus resultados de busca retornarão na funcionalidade de busca por GPS. Além disso, é nela que o aplicativo captura as coordenadas geográficas atuais do dispositivo para descobrir qual o município que o usuário se encontra, no entanto, o mesmo também pode definir mais municípios de interesse que serão utilizados na funcionalidade de busca por parâmetros. A Figura 5 apresenta a tela descrita.

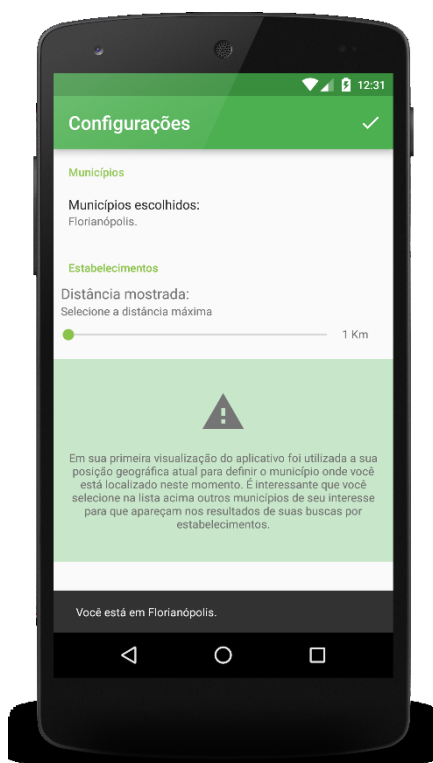


Figura 46. Tela de configuração inicial

Fonte: Desenvolvido pelo autor

Após todas as configurações serem feitas o aplicativo está preparado para fornecer todas as funcionalidades propostas. A seção seguinte mostra a tela inicial do aplicativo onde o usuário poderá visualizar todos os seus estabelecimentos favoritos.

4.2.3. A tela inicial do aplicativo

A tela inicial do aplicativo permite que o usuário acesse o menu de opções, esse fornece acesso a algumas funcionalidades como: visualizar informações sobre o aplicativo; visualiza a opção que dá acesso a tela de ajuda, que auxilia o usuário listando e explanando os vários tipos de estabelecimentos de saúde e de serviços especializados existentes, explicando alguns termos técnicos; visualizar a opção que fornece acesso a tela de configuração. A Figura 6 mostra as opções possíveis.

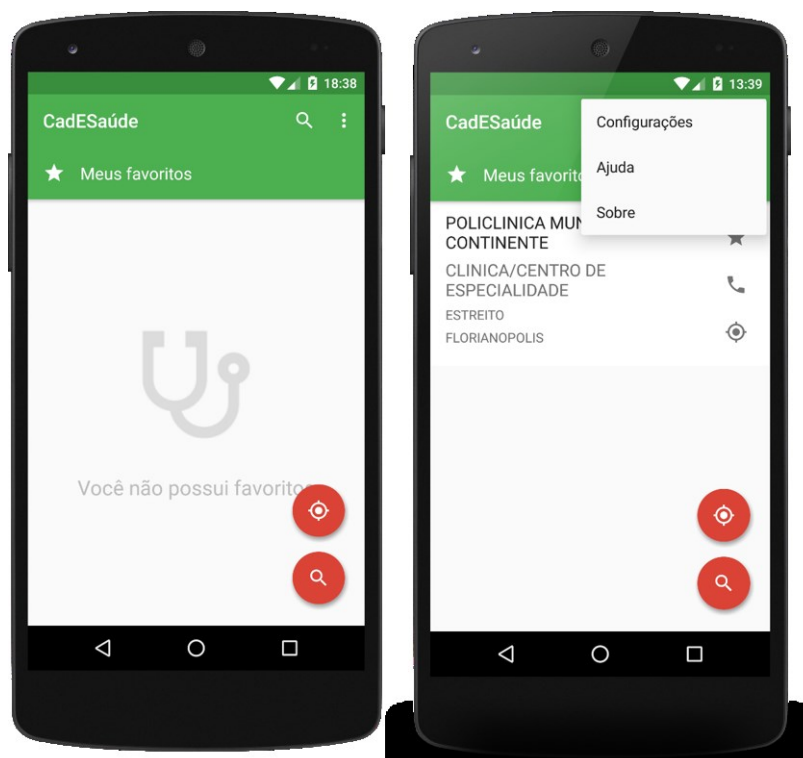


Figura 47. Tela inicial

Fonte: Desenvolvido pelo autor

A partir desta tela o usuário poderá ter acesso as seguintes funcionalidades:

- Busca de estabelecimentos por localização geográfica. Para tal ele deverá acionar o botão de ação flutuante com símbolo de GPS;
- Busca de estabelecimentos por parâmetros. É possível acessar esta funcionalidade tocando no botão de ação flutuante com ícone de lupa.
- Esta tela apresenta uma lista de estabelecimentos contendo os estabelecimentos que foram definidos como favoritos em uma situação anterior. Pode-se interagir com esta lista e executar ações com cada um dos estabelecimentos contidos nela. É possível visualizar o cadastro completo do estabelecimento tocando nesse item na lista. Pode-se retirá-lo dos favoritos tocando no ícone de estrela. Há a possibilidade de ligar para o estabelecimento tocando no ícone do telefone ou visualizar sua posição em um mapa tocando no ícone de GPS.
- O usuário pode utilizar a busca pelo nome de um estabelecimento que esteja em sua lista de favoritos, basta o mesmo tocar no ícone de lupa na parte superior da tela, na barra de ações para que um campo de busca seja habilitado e se possa introduzir o termo a ser buscado.

Esta tela é a principal do aplicativo. Para fechá-lo, basta o pressionar o botão *back* do dispositivo.

4.2.4. A tela de visualização do estabelecimento

A tela que detalha todas as informações sobre um estabelecimento pode ser vista na Figura 7. Nela pode-se obter informações sobre sua localização e telefone de contato, assim como os profissionais vinculados, os serviços que ela possui, as formas de atendimentos (emergencial, ambulatorial, internação) e seus tipos de convênios aceitos (SUS, particular, Convênio).

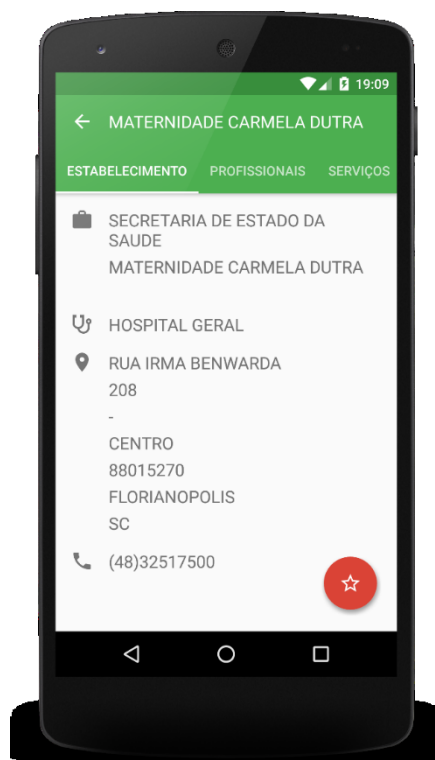


Figura 48. Tela de detalhes do estabelecimento

Fonte: Desenvolvido pelo autor

Nesta tela o usuário também tem acesso as funcionalidades já informadas anteriormente: adicionar ou excluir o estabelecimento aos seus favoritos tocando no botão com ícone de estrela, ligar para o estabelecimento tocando no ícone de telefone e ver o estabelecimento em um mapa tocando no ícone de marcação de local.

Todos os dados mostrados nesta tela são dados que estão armazenados no banco de dados interno do aplicativo. O mesmo banco gerado com a ferramenta de captura de dados explicada na seção 4.1. Todos os processamentos das consultas são feitas utilizando os recursos do dispositivo, logo, ele não necessita de uma conexão com a internet para funcionar, exceto quando a funcionalidade de visualizar a localização do estabelecimento em um mapa, pois, para esta funcionalidade, o aplicativo repassa essa solicitação a plataforma do Android e ela solicita tal ação ao aplicativo padrão de visualização de mapas do dispositivo.

4.2.5. A tela de busca por GPS

A tela que possui a principal funcionalidade do aplicativo pode ser vista na Figura 8. Esta tela de busca de estabelecimentos utiliza o valor definido na tela de configurações explicado na seção 4.2.2. Esse valor referente ao raio de busca é exibido no subtítulo da tela para que o usuário seja informado sobre a distância máxima dos resultados. Ao entrar nesta tela, a qual é acessível através da tela inicial como visto na seção 4.2.3, um processo em segundo plano faz o processamento dos estabelecimentos armazenado no banco de dados que estão dentro do valor do raio de distância pré-definido. Esses estabelecimentos são apresentados na tela em uma lista de itens que possuem as mesmas funcionalidades apresentadas na seção 4.2.3.

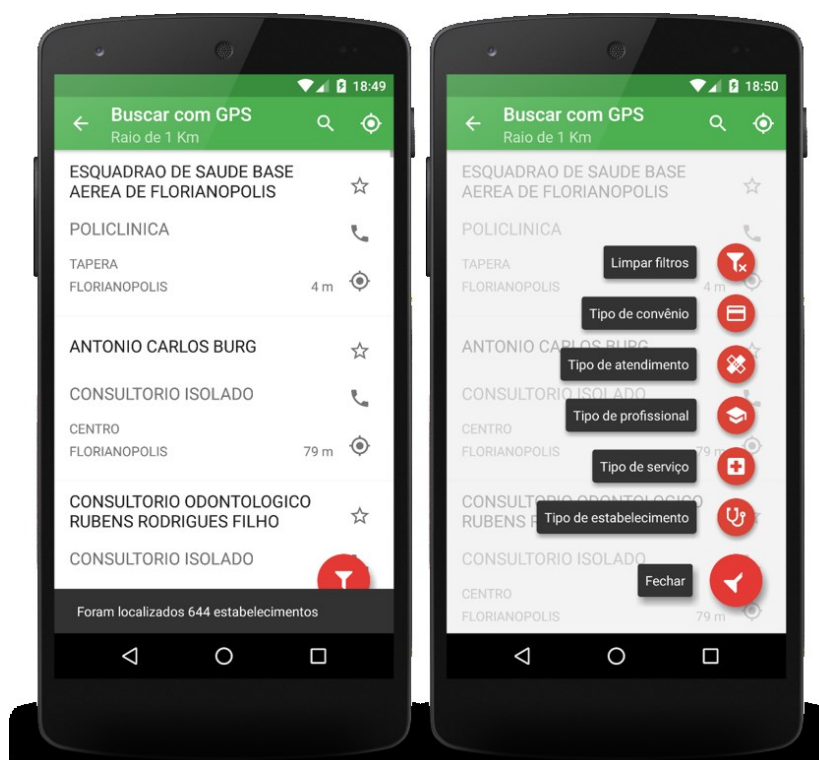


Figura 49. Tela de busca por GPS

Fonte: Desenvolvido pelo autor

A tela de buscas ainda permite que uma série de filtros sejam utilizados para ajudar ao usuário na busca pelo estabelecimento que atenda suas necessidades. Existe ainda a possibilidade de uma busca pelo nome do estabelecimento, essa é feita tocando no ícone de lupa localizado na barra de ações no topo da tela. O usuário tem a possibilidade de reprocessar as buscas e calcular as distâncias tocando no ícone de GPS localizado na barra de ferramentas, no topo da tela.

4.2.6. A tela de busca por parâmetros

Esta funcionalidade permite que o usuário faça a busca por estabelecimentos de saúde localizados dentro dos municípios que foram escolhidos na tela de configurações do aplicativo. Este processo foi explanado na seção 4.2.2, sendo que esse influencia nos resultados das buscas realizadas no banco de dados, sendo essa realizada por meio da seleção de alguma característica específica do estabelecimento como parâmetro. A Figura 9 demonstra as telas onde são feitas as configurações das buscas.

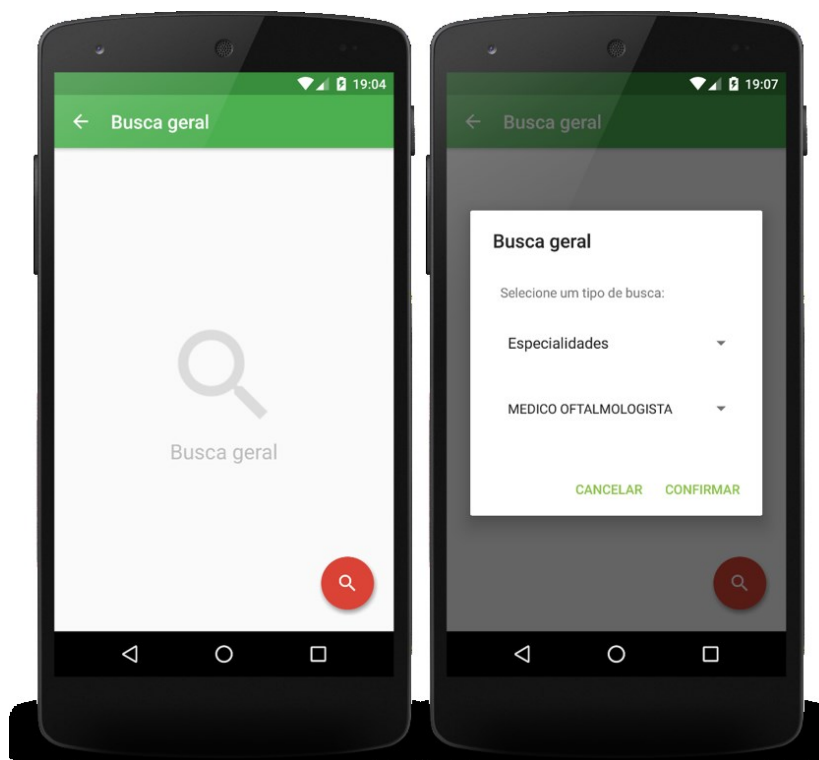


Figura 50. Tela de busca por parâmetros

Fonte: Desenvolvido pelo autor

A busca a ser realizada é configurada em uma janela de diálogo. Esta janela apresenta os possíveis valores para parâmetros de busca baseados nas características dos estabelecimentos localizados nos municípios que o usuário configurou no aplicativo, como foi explicado na seção 4.2.2. Ao montar a configuração da busca e confirma-la, a janela se fecha e uma lista é apresentada contendo os resultados das buscas, essa lista possui as mesmas características e funcionalidades da lista apresentada na seção 4.2.3. Para realizar uma nova busca o usuário deverá tocar no botão com ícone de lupa para que a janela de diálogo seja apresentada novamente.

4.2.7. As atualizações do aplicativo

O aplicativo utiliza como base para as consultas os dados referentes ao CNES, os quais são atualizados mensalmente pelas Secretarias Municipais e Estaduais de Saúde, de acordo com as legislações que regulam esse processo (BRASIL, 2015). Como a manutenção desses cadastros é feita mensalmente, pensou-se em lançar uma nova versão do aplicativo a cada mês, contendo os dados recentes sobre os estabelecimentos. Os usuários que possuem alguma versão instalada serão notificados sobre as novas versões lançadas, visto que o aplicativo se encontra disponível na loja de aplicativo da Google.

5. Conclusão

Com o desenvolvimento da ferramenta de captura de dados, a qual gera o banco de dados com as informações recentemente atualizadas dos estabelecimentos de saúde que se encontram no CNES, juntamente com o aplicativo para a plataforma Android, foi possível atingir o principal objetivo deste trabalho. Após seu término, acrescenta-se na loja de aplicativos da Google mais uma ferramenta de utilidade pública para a população. Espera-se que ela faça parte da rotina do cidadão do nosso Estado e de seus visitantes em sua estadia em algum município de Santa Catarina.

Para trabalhos futuros, sugere-se uma análise da usabilidade do aplicativo, pois não foi possível cobrir esta questão durante o desenvolvimento do sistema, sendo assim, levanta-se aqui a sugestão ao leitor para o desenvolvimento dos mais atuais testes de usabilidade com usuários reais e através dos resultados de uma análise aprofundada, caso haja necessidade, providencie a reengenharia da interface gráfica do aplicativo para que a mesma se torne mais intuitiva.

Seria de grande utilidade se todos os Estados do país pudessem ter acesso a um aplicativo como este, porém a forma como foi desenvolvido, sendo um aplicativo totalmente off-line, não suportaria um banco de dados contendo todos os estabelecimentos do país, visto que o tamanho do seu arquivo poderia ultrapassar os limites definidos pela plataforma do Android, além disso, os recursos de processamento teriam que ser maiores e aumentaria o consumo do dispositivo. Sendo assim sugere-se para trabalhos futuros um estudo com maior aprofundamento, afim de promover a melhor solução para disponibilizar a toda população do país uma ferramenta que auxilie na localização de estabelecimentos de saúde.

Referências

- BRASIL. MINISTRO DE ESTADO DA SAÚDE. Portaria nº 1.646, de 02 de outubro de 2015. Diário Oficial da União, Brasília, 2 out. 2015, v. , n. 190, p. 669-670.
- EXAME.COM. Vendas mundiais de smartphones crescem 23% em 2014. Disponível em: <<http://exame.abril.com.br/tecnologia/noticias/vendas-mundiais-de-smartphones-crescem-23-em-2014>>. Acesso em: 23 mai. 2015.
- CLINICA PRIVADA DE ESPECIALIDADES VILLA MARÍA. Clinica Especialidades. Disponível em: <<https://play.google.com/store/apps/details?id=clinicaespecialidades.myapp.name>>. Acesso em: 25 out. 2015.
- PORTAL UNIMED LTDA. Guia Médico Nacional Unimed para smartphone. Disponível em: <<https://play.google.com/store/apps/details?id=com.unimed.guiamdico>>. Acesso em: 23 mai. 2015.
- WINTIMEIPC. MedSquare. Disponível em: <https://play.google.com/store/apps/details?id=br.com.medsquare.medsquare&hl=pt_BR>. Acesso em: 24 mai. 2015.
- TAVARES, Alberto Trindade; LÓSCIO, Bernadette Farias. Processamento de consultas na Web de Dados: uma abordagem para busca de fontes de dados relevantes. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS. 28., 2013, Recife. Short Papers... Recife: SBC, 2013. Disponível em: <<http://sbbd2013.cin.ufpe.br/Proceedings/artigos/pdfs/SBBD-Proceedings-Complete.pdf>>. Acesso em: 25 out. 2015.
- TAVARES, Alberto Trindade; LÓSCIO, Bernadette Farias; OLIVEIRA, Hélio Rodrigues de. Buscando Fontes de Dados Relevantes para Aplicações Linked Data. In: SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E WEB. 18., 2012, São Paulo. Anais IX Workshop de Trabalhos de Iniciação Científica... São Paulo: SBC, 2012. Disponível em: <http://sws2012.ime.usp.br/download/WebMedia2012_Anais.pdf>. Acesso em: 25 out. 2015.
- TABORDA, Sérgio Manuel Marcos. DAO. Disponível em: <<https://sergiotaborda.wordpress.com/desenvolvimento-de-software/java/patterns/dao>>. Acesso em: 30 mai. 2015.
- FREEMAN, Eric; FREEMAN, Elisabeth; SIERRA, Kathy; et al. Use a Cabeça: Padrões e Projetos. 2 ed. Rio de Janeiro: Alta Books, 2009.

BRASIL. MS-SAS: Secretaria de Assistência a Saúde . Portaria nº 511, de 29 de dezembro de 2000. Diário Oficial da União, Brasília, 4 jan. 2001, v. , n. 3-E, p. 11-20.

BRASIL. MINISTRO DE ESTADO DA SAÚDE. Portaria nº 1.646, de 02 de outubro de 2015. Diário Oficial da União, Brasília, 2 out. 2015, v. , n. 190, p. 669-670.

MEYER, Maximiliano. A história do Android. Disponível em: <<https://www.oficinadanet.com.br/post/13939-a-historia-do-android>>. Acesso em: 31 ago. 2016.

ALVES, Paulo. A evolução do Android: do Cupcake ao Marshmallow, conheça todas as versões. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2015/10/a-evolucao-do-android-do-cupcake-ao-marshmallow-conheca-todas-as-versoes.html>>. Acesso em: 31 ago. 2016.

ANDROID. Platform Architecture. 2016. Disponível em: <<https://developer.android.com/guide/platform/index.html>>. Acesso em: 1 set. 2016.

SARINHO, Victor Travassos; CAMPOS, Luan Rios. SAMob - Sistema Móvel de Geolocalização e Geoprocessamento para Locais de Atendimento em Saúde. Journal of Health Informatics. São Paulo; v. 6, n. 3, set. 2014. p. 96-101. Disponível em: <<http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/view/304>>. Acesso em: 18 out. 2015.

MEDSQUARE. medsquare. Disponível em: <<http://www.medsquare.com.br>>. Acesso em: 25 out. 2015.

ANEXOS

ANEXO A - TIPOS DE ESTABELECIMENTOS/UNIDADES

01 - Posto de Saúde: Unidade destinada à prestação de assistência a uma determinada população, de forma programada ou não, por profissional de nível médio, com a presença intermitente ou não do profissional médico.

02 - Centro de Saúde/Unidade Básica de Saúde: Unidade para realização de atendimentos de atenção básica e integral a uma população, de forma programada ou não, nas especialidades básicas, podendo oferecer assistência odontológica e de outros profissionais de nível superior. A assistência deve ser permanente e prestada por médico generalista ou especialista nestas áreas. Podendo ou não oferecer: SADT e Pronto atendimento 24 Horas.

04 - Policlínica: Unidade de saúde para prestação de atendimento ambulatorial em várias especialidades, incluindo ou não as especialidades básicas, podendo ainda ofertar outras especialidades não médicas. Podendo ou não oferecer: SADT e Pronto atendimento 24 Horas.

05 - Hospital Geral: Hospital destinado à prestação de atendimento nas especialidades básicas, por especialistas e/ou outras especialidades médicas. Pode dispor de serviço de Urgência/Emergência. Deve dispor também de SADT de média complexidade. Podendo ter ou não SIPAC.

07 - Hospital Especializado: Hospital destinado à prestação de assistência à saúde em uma única especialidade/área. Pode dispor de serviço de Urgência/Emergência e SADT. Podendo ter ou não SIPAC Geralmente de referência regional, macrorregional ou estadual.

15 - Unidade Mista: Unidade de saúde básica destinada à prestação de atendimento em atenção básica e integral à saúde, de forma programada ou não, nas especialidades básicas, podendo oferecer assistência odontológica e de outros profissionais, com unidade de internação, sob administração única. A assistência médica deve ser permanente e prestada por médico especialista ou generalista. Pode

dispor de urgência/emergência e SADT básico ou de rotina. Geralmente nível hierárquico 5.

20 - Pronto Socorro Geral: Unidade destinada à prestação de assistência a pacientes com ou sem risco de vida, cujos agravos necessitam de atendimento imediato. Podendo ter ou não internação.

21 - Pronto Socorro Especializado: Unidade destinada à prestação de assistência em uma ou mais especialidades, a pacientes com ou sem risco de vida, cujos agravos necessitam de atendimento imediato. MS/SAS/DRAC/CGSI-Coordenação Geral dos Sistemas de Informação Manual Técnico do Cadastro Nacional de Estabelecimentos de Saúde - Versão 2 Atualizado em outubro de 2006.

22-Consultório Isolado: sala isolada destinada à prestação de assistência médica ou odontológica ou de outros profissionais de saúde de nível superior.

32-Unidade Móvel Fluvial: Barco/navio equipado como unidade de saúde, contendo no mínimo um consultório médico e uma sala de curativos, podendo ter consultório odontológico.

36-Clínica Especializada/Amb. Especializado: Clínica Especializada destinada à assistência ambulatorial em apenas uma especialidade/área da assistência. (Centro Psicossocial/Reabilitação, etc..).

39 - Unidade de Serviço de Apoio de Diagnose e Terapia: Unidades isoladas onde são realizadas atividades que auxiliam a determinação de diagnóstico e/ou complementam o tratamento e a reabilitação do paciente.

40 - Unidade Móvel Terrestre: Veículo automotor equipado, especificamente, para prestação de atendimento ao paciente.

42 - Unidade Móvel de Nível Pré-hospitalar na Área de Urgência e Emergência: Veículo terrestre, aéreo ou hidroviário destinado a prestar atendimento de urgência e

emergência pré-hospitalar a paciente vítima de agravos a sua saúde. (PTMS/GM 824, de 24/Jun/1999).

43 - Farmácia: Estabelecimento de saúde isolado em que é feita a dispensação de medicamentos básicos/essenciais (Programa Farmácia Popular) ou medicamentos excepcionais/alto custo previstos na Política Nacional de Assistência Farmacêutica.

50 - Unidade de Vigilância em Saúde: É o estabelecimento isolado que realiza trabalho de campo a partir de casos notificados e seus contatos, tendo como objetivos: identificar fontes e modo de transmissão; grupos expostos a maior risco; fatores determinantes; confirmar o diagnóstico e determinar as principais características epidemiológicas, orientando medidas de prevenção e controle a fim de impedir a ocorrência de novos eventos e/ou o estabelecimento de saúde isolado responsável pela execução de um conjunto de ações, capaz de eliminar, diminuir ou prevenir riscos à saúde capaz de eliminar, diminuir ou prevenir riscos à saúde e de intervir nos problemas sanitários decorrentes do meio ambiente, da produção e circulação de bens e da prestação de serviços de interesse da saúde.

60 - Cooperativa: Unidade administrativa que disponibiliza seus profissionais cooperados para prestarem atendimento em estabelecimento de saúde.

61 - Centro de Parto Normal Isolado: Unidade intra-hospitalar ou isolada, especializada no atendimento da mulher no período gravídico puerperal, conforme especificações da PT/MS 985/99.

62 - Hospital /Dia- Isolado: – Unidades especializadas no atendimento de curta duração com caráter intermediário entre a assistência ambulatorial e a internação.

64 - Central de Regulação de Serviços de Saúde: É a unidade responsável pela avaliação, processamento e agendamento das solicitações de atendimento, garantindo o acesso dos usuários do SUS, mediante um planejamento de referência e contra referência.

67 - Laboratório Central de Saúde Pública - LACEN- Estabelecimento de Saúde que integra o Sistema Nacional de Laboratórios de Saúde Pública - SISLAB, em conformidade com normalização vigente.

ANEXO B - TIPOS DE SERVIÇOS ESPECIALIZADOS

Código	Descrição do serviço
100	Serviço de Atenção à Saúde no Sistema Penitenciário
101	Estratégia de Saúde da Família
102	Estratégias de Agentes Comunitário de Saúde
103	Serviço de Atendimento Móvel de Urgências
104	Regulação Assistencial dos Serviços de Saúde
105	Serviço de Atenção em Neurologia/ Neurocirurgia
106	Serviço de Atenção a DST/HIV/AIDS
107	Serviço de Atenção à Saúde Auditiva
108	Serviço de Atenção a Saúde do Trabalhador
109	Serviço de Atenção à Saúde dos Adolescentes em Conflito com a Lei, em Regime de Internação e Internação Provisória
110	Serviço de Atenção à Saúde Reprodutiva
111	Serviço de Atenção ao Paciente com Tuberculose
112	Serviço de Atenção ao Pré-Natal, Parto e Nascimento
113	Serviço de Atenção Domiciliar
114	Serviço de Atenção em Saúde Bucal
115	Serviço de Atenção Psicossocial
116	Serviço de Atenção Cardiovascular/Cardiologia
117	Serviço de Cirurgia Reparadora
118	Serviço de Cirurgia Torácica
119	Serviço de Controle de Tabagismo
120	Serviço de Diagnóstico por Anatomia Patológica e/ou Cito patologia
121	Diagnóstico por Imagem
122	Serviço de Diagnóstico por Métodos Gráficos/Dinâmicos
123	Serviço de Dispensação de Órteses, Próteses e Materiais Especiais
124	Serviço de Endocrinologia
125	Serviço de Farmácia
126	Serviço de Fisioterapia
127	Serviço de Atenção à Obesidade
128	Serviço de Hemoterapia
129	Serviço de Laboratório de Histocompatibilidade
130	Atenção à Doença Renal Crônica
131	Serviço de Oftalmologia
132	Serviço de Oncologia
133	Serviço de Pneumologia
134	Serviço de Práticas Integrativas e Complementares
135	Serviço de Reabilitação
136	Serviço de Suporte Nutricional
137	Serviço de Telemedicina
138	Serviço de Traumato-Ortopedia
139	Serviço de Triagem Neonatal
140	Serviço de Urgência e Emergência
141	Serviço de Vigilância em Saúde

142	Serviço de Endoscopia
143	Serviço Móvel de Urgência (exceto SAMU)
144	Serviço Posto de Coleta de Materiais Biológicos
145	Serviço de Diagnóstico por Laboratório Clínico
146	Serviço de Videolaparoscopia
147	Serviço de Apoio a Saúde da Família
148	Hospital-Dia
149	Transplante
150	Cirurgia Vascular
151	Medicina Nuclear
153	Atenção Especializada no Processo Transexualizador
154	Serviço de Banco de Tecidos
155	Serviço de Traumatologia Ortopedia
156	Serviço de Atenção à Saúde das Pessoas Ostomizadas
157	Serviço de Laboratório de Prótese Dentaria
158	Serviço de Atenção Integral em Hanseníase
159	Atenção Básica
160	Tele consultoria
162	Serviço de Terapia Intensiva
163	Serviço de Cuidados Intermediários
164	Serviço de Órteses, Próteses e Materiais Especiais em Reabilitação
165	Atenção às Pessoas em Situação de Violência Sexual
167	Atendimento Itinerante de Assistência e Ensino em Saúde
168	Atenção às Pessoas com Doenças Raras
169	Serviço de Atenção em Urologia